РУКОВОДСТВО ПО МИГРАЦИИ НА LINUX

ДЛЯ РЕГИОНАЛЬНЫХ АДМИНИСТРАЦИЙ

Санкт-Петербург, 2006

Copyright © IBM Corp. 2004. All rights reserved.
Copyright © ОАО Линукс Инк, 2006 (перевод, макет, часть I, приложения E, F)

Реферат

Документ содержит методические рекомендации по анализу ИТ-инфраструктуры региональной администрации и определению перспективных направлений ее модернизации на основе программного обеспечения с открытым кодом. Сформулированы требования к компонентам ИТ-инфраструктуры.

Данное руководство основывается на материалах, представленнных в издании IBM RedBook *Linux Client Migration Cookbook, A Practical Planning and Implementation Guide for Migrating to Desktop Linux*:

http://www.redbooks.ibm.com/abstracts/sg246380.html

В работе использованы результаты, полученные компанией «ОАО Линукс Инк», см.:

http://www.linux-ink.ru

в ходе проведения работ по созданию концепции модернизации ИТ-инфраструктуры органов управления Новгородской области (проект выполнялся при поддержке Министерства Экономического Развития и Торговли), а также ряда других проектов, выполнявшихся этой компанией.

СОДЕРЖАНИЕ

Введение 11

I Модернизация ИКТ-инфраструктуры 27

1 Текущее состояние информационных систем 29
1.1 Обеспечение доступа граждан к государственным информационным ресурсам . . . 29
1.2 Структура информационного обмена . . . 29
1.3 Используемое программное обеспечение . . . 30
1.4 Подготовка кадров . . . 30
1.5 Текущее состояние ИТ-инфраструктур . . . 31

2 Модернизация ИТ-инфраструктуры 33
2.1 Базовое программное обеспечение (БПО) . . . 35
2.2 Информационно-коммуникационные средства . . . 36
2.3 Пользовательское ПО общего назначения . . . 37
2.4 Инфраструктура рабочих мест . . . 37
2.5 Системы электронного документооборота . . . 39
2.6 Информационные порталы . . . 39
2.7 Геоинформационные системы . . . 40
2.8 Применение GRID-технологий . . . 43

3 Модернизация системной инфраструктуры 45
3.1 Центр Обработки Данных (ЦОД) . . . 45
3.2 Модернизация БПТИ . . . 46
3.2.1 Операционные системы . . . 49
3.2.2 Утилиты системного и общего назначения . . . 49
3.2.3 Серверная инфраструктура . . . 49
3.2.4 Средства администрирования программно-аппаратного комплекса . . . 50
3.2.5 Технологическая платформа интеграции, быстрой разработки приложений . . . 50
3.2.6 Размещение компонентов БПТИ и организация удаленных сегментов . . . 50
3.3 Организация рабочего окружения пользователя (РОП) . . . 51
3.4 Геоинформационная система (ГИС) . . . 52
3.5 Преимущества и особенности предлагаемых решений . . . 52

II Выбор Linux 55

4 Миграция на Linux 57
4.1 Как обстоит дело с миграцией сегодня . . . 57
4.2 Поиск оптимального решения . . . 58
4.3 Стратегическое планирование . . . 58
4.4 Планирование на уровне клиентов . . . 59
4.5 Чем хорош Linux . . . 60
4.6 Обзор Linux и вариантов дистрибутивов . . . 60
4.7 Будущее настольных систем под Linux . . . 61
4.7.1 От серверов к настольным системам: Linux Standard Base привлекает разработчиков программного обеспечения по всему миру . . . 62
4.7.2 Рабочий стол под Linux сегодня . . . 62
4.7.3 Состав Linux Standards Base . . . 62
4.7.4 Стимулы для независимых производителей программных продуктов . . . 63
4.7.5 Существенное увеличение числа потенциальных пользователей . . . 64
4.7.6 LSB-совместимые дистрибутивы . . . 64
4.7.7 LSB-совместимые приложения . . . 64

4.7.8 Конечные пользователи . . . 65
4.7.9 Вехи на пути к широкомасштабному внедрению . . . 65
4.7.10 LSB Desktop Project . . . 65

5 Аргументы в пользу миграции 67
5.1 Зачем мигрировать . . . 67
5.1.1 Безопасность персонального компьютера . . . 68
5.1.1.1 Безопасность броузера . . . 68
5.1.1.2 Безопасность клиента электронных сообщений . . . 68
5.1.1.3 Защита пользователя и обеспечение безопасности . . . 69
5.1.1.4 Оперативность исправления ошибок . . . 69
5.1.1.5 Брандмауэр . . . 69
5.1.2 Стоимость рабочего места под Linux . . . 69
5.1.2.1 Стоимость лицензии и поддержки дистрибутива Linux . . . 70
5.1.2.2 Стоимость оборудования . . . 70
5.1.2.3 Стоимость базового набора приложений клиентской машины . . . 70
5.1.2.4 Затраты на сопровождение и поддержку . . . 70
5.1.3 Уровень управляемости Linux клиента . . . 70
5.1.3.1 Модульная структура операционной системы . . . 71
5.1.3.2 Механизм обновлений . . . 71
5.1.3.3 Удаленный доступ . . . 71
5.1.3.4 Инструменты для удаленного управления . . . 72
5.1.4 Настройка системы под нужды пользователя . . . 72
5.1.4.1 Гибкость в добавлении или удалении компонент . . . 73
5.1.4.2 Предотвращение «разбухания» клиента . . . 73
5.1.4.3 Создание дистрибутивов, ориентированных на конкретные задачи . . . 73
5.1.4.4 Настройка рабочего стола . . . 73
5.1.5 Философия открытого ПО . . . 73
5.1.6 Простота в эксплуатации . . . 73
5.2 Когда следует мигрировать, а когда нет . . . 73
5.2.1 Функции на клиентской машине соответствуют модели «Тонкий» или модели «Стройный» . . . 74
5.2.2 Высокое число мигрируемых приложений . . . 74
5.2.3 Организационная готовность . . . 74
5.3 В чем заключаются особенности Linux? . . . 75
5.3.1 Много пользователей — зачем? . . . 75
5.3.2 Что такое ядро? . . . 76
5.3.3 Взаимодействие приложений или одно большое приложение? . . . 76
5.3.4 Пользовательский интерфейс, на что обратить внимание? . . . 76
5.3.5 Как настольная система взаимодействует с внешней инфраструктурой? . . . 76
5.3.6 Каким образом хранятся данные? . . . 77
5.3.7 Есть ли другие отличия? . . . 77
5.4 Цели миграции . . . 77
5.4.1 Пилотная миграция . . . 78
5.4.2 Полная миграция . . . 78

III Планирование пилотной миграции 79

6 Организационное планирование и человеческий фактор 81
6.1 Изучение паттернов использования компьютеров . . . 81
6.1.1 Разделение рабочих мест на функциональные группы . . . 81
6.1.2 Описание профилей рабочих мест . . . 83
6.1.3 Опрос конечных пользователей . . . 83
6.2 Обеспечение функциональной преемственности . . . 83
6.2.1 Приложения, работающие на обеих платформах . . . 83
6.2.2 Функциональные эквиваленты служебных приложений . . . 84
6.2.3 Веб-приложения . . . 85
6.2.4 Сервер-ориентированная обработка данных . . . 85
6.3 Человеческий фактор . . . 86

6.4 Вопросы переобучения . . . 87
6.4.1 Кросс-платформенные приложения помогут отделить переобучение от миграции . . . 87
6.4.2 Сохраняем визуальный интерфейс . . . 87
6.4.3 Привычные действия . . . 87
6.4.4 Файловая система: сохранить структуру подкаталогов . . . 87
6.4.5 До миграции: просто попробуем поработать с Linux . . . 87

7 Техническое планирование **89**
7.1 Изучение имеющейся информационной среды . . . 89
7.1.1 Оценка общей ситуации . . . 90
7.1.2 Инвентаризация оборудования . . . 90
7.1.3 Инвентаризация программного обеспечения . . . 91
7.1.4 Описание форматов данных и их взаимозависимостей . . . 91
7.1.5 Описание сетевой инфраструктуры . . . 92
7.1.6 Описание функций конечных пользователей . . . 92
7.2 Взаимодействие с имеющимися сетевыми службами . . . 94
7.2.1 Различные типы сетевого окружения . . . 94
7.2.2 Аутентификация в домене Windows . . . 95
7.2.3 Использование средств домена для совместного доступа к файлам . . . 96
7.2.3.1 Домашние каталоги и разделяемые ресурсы . . . 96
7.2.4 Использование сетевых принтеров . . . 97
7.2.4.1 Взаимодействие CUPS и Samba . . . 97
7.2.4.2 Принтеры и аутентификация . . . 97
7.2.4.3 Принт-сервер или прямая печать? . . . 97
7.2.4.4 Поддерживает ли CUPS модель/драйвер вашего принтера? . . . 98
7.2.5 DHCP и DNS-конфигурация . . . 98
7.2.6 Доступ через прокси-сервер . . . 98
7.3 Стандартизация рабочего стола . . . 98
7.3.1 Дистрибутивы Linux . . . 99
7.3.2 Менеджеры рабочего стола Linux . . . 99
7.3.2.1 Менеджер рабочего стола KDE . . . 100
7.3.2.2 Менеджер рабочего стола GNOME . . . 101
7.3.3 Внешний вид и функциональность . . . 101
7.3.3.1 Иконки на рабочем столе . . . 101
7.3.3.2 Темы рабочего стола . . . 102
7.3.3.3 Дизайн меню приложений . . . 102
7.3.4 Ограничения для пользователей . . . 104
7.3.5 Выбор приложений . . . 104
7.3.6 Файловые системы и разделы . . . 105
7.4 Планирование миграции приложений . . . 105
7.4.1 Возвращение к архитектуре клиент/сервер . . . 105
7.4.2 Разделение клиентов «по весу» — Тонкий, Стройный и Толстый . . . 105
7.4.3 Системные решения для Linux . . . 106
7.5 Планирование управления Linux-клиентом . . . 106
7.5.1 Обновления и изменения операционной системы . . . 107
7.5.2 Обновление приложений . . . 108
7.5.3 Удаленное управление клиентами . . . 109
7.5.4 Процедура добавления или замены клиентов . . . 109
7.5.5 Резервные копии клиентских рабочих мест . . . 109
7.5.6 Антивирусные мероприятия . . . 110
7.5.7 Службы администрирования в промышленных дистрибутивах . . . 111
7.5.7.1 Администрирование в дистрибутиве Red Hat Desktop . . . 111
7.5.7.2 Администрирование в дистрибутиве Novell Linux Desktop . . . 112
7.6 Сравнение процедуры миграции настольных машин и ноутбуков . . . 113
7.6.1 Аппаратные средства . . . 114
7.6.2 Периферийные устройства . . . 115
7.6.3 Соединение с интернетом . . . 115
7.6.4 Работа в автономном режиме . . . 116
7.6.4.1 Автономная работа почтового клиента . . . 116

7.6.4.2 Доступ к файлам в автономном режиме . . . 116
7.7 Приложения, не допускающие переноса под Linux . . . 117
7.7.1 Что мешает переносу приложений под Linux . . . 117
7.7.2 Использование Terminal Server, Citrix Metaframe или NoMachine NX . . . 117
7.7.3 Использование VMware . . . 118
7.7.4 Использование возможности альтернативной загрузки . . . 118
7.7.5 Что делать, если ни один из вариантов не подходит . . . 119
7.8 Развертывание новых клиентов . . . 119
7.8.1 Метод развертывания новых клиентов . . . 119
7.8.1.1 Метод установки на базе готового образа . . . 120
7.8.1.2 Метод автоматической установки с дистрибутива . . . 120
7.8.2 Технология обновления клиентского ПО . . . 120
7.8.3 Персональные настройки новых клиентов . . . 120
7.9 Выявление неисправностей и техническая поддержка после миграции . . . 121
7.9.1 Каких сложностей следует ожидать . . . 121
7.9.2 Как справляться с неожиданными сложностями . . . 122
7.9.3 Когда обращаться в службу поддержки производителя ПО . . . 122

8 Архитектура Linux и технические особенности 123
8.1 Что такое Linux? . . . 123
8.1.1 Свободное программное обеспечение и программное обеспечение с открытым кодом . . . 124
8.1.2 Дистрибутивы . . . 124
8.1.3 Стандарты . . . 125
8.2 Технические отличия . . . 126
8.2.1 Процесс загрузки ядра . . . 126
8.2.2 Внешние взаимодействия, файлы и сервисы . . . 127
8.2.3 Многопользовательский режим . . . 128
8.2.3.1 Могут ли несколько пользователей запускать процессы одновременно? . . . 129
8.2.3.2 Могут ли несколько пользователей открыть свою сессию на компьютере одновременно? . . . 129
8.2.3.3 Может ли быть установлено приложение, допускающее одновременное использование несколькими пользователями? . . . 129
8.2.3.4 Как устанавливаются полномочия для различных пользователей? . . . 129
8.2.4 Графическое окружение и текстовый режим . . . 130
8.2.5 Системные уровни выполнения . . . 132
8.2.6 Диски, разделы и файловые системы . . . 132
8.2.7 Виртуальная память . . . 133
8.2.8 Ссылки . . . 134
8.2.9 Замена ядра . . . 134
8.2.10 Драйверы устройств и поддержка аппаратного обеспечения . . . 135
8.2.11 Поддержка шрифтов . . . 135
8.2.12 Поддержка 64-разрядных и многоядерных процессоров . . . 135

IV Пилотная миграция 137

9 Модели развертывания клиентов 139
9.1 Среда KDE Kiosk . . . 139
9.1.1 Пользовательские профили . . . 139
9.1.2 Опции блокировки . . . 142
9.1.3 Неизменяемые элементы конфигурационных файлов . . . 145
9.1.4 Ограничения на действия . . . 145
9.1.5 Ограничения на URL . . . 147
9.1.6 Ограничения на ресурсы . . . 147
9.1.7 Ограничения в среде KDE Kiosk . . . 149
9.1.8 Подведем итоги . . . 149
9.2 Конфигурирование GNOME . . . 149
9.2.1 Конфигурация в GNOME . . . 149
9.2.2 Инструмент конфигурирования gconftool-2 . . . 150

9.2.3 Ограничения в рабочем окружении пользователя . . . 152
9.2.4 Настройка внешнего вида рабочего стола . . . 153
9.3 Инструменты удаленного доступа . . . 154
9.3.1 Удаленный доступ . . . 154
9.3.2 Тонкий клиент . . . 154
9.3.3 Перемещение приложений . . . 155
9.3.4 Мультитерминальная обработка данных . . . 155

10 Сценарий миграции клиента **157**
10.1 Пример миграции клиента . . . 157
10.1.1 Описание паттернов использования клиента . . . 157
10.1.2 Определение наиболее существенных приложений и потребностей в интеграции . . . 157
10.2 Детальный план миграции . . . 158
10.2.1 Какого типа этот клиент? . . . 158
10.2.2 Какое выбрать графическое окружение? . . . 158
10.2.3 Какое аппаратное обеспечение используется? . . . 158
10.2.4 План замены программного обеспечения . . . 158
10.2.5 Подключение к сети Windows . . . 159
10.3 Процесс миграции . . . 159
10.3.1 Выполнение базовой инсталляции . . . 160
10.3.1.1 Установление удаленного доступа администратора с использованием VNC . . . 160
10.3.1.2 Подготовка к интеграции с сетевыми сервисами Windows . . . 160
10.3.2 Интеграция с существующими сетевыми сервисами . . . 161
10.3.2.1 Монтирование общедоступных каталогов Windows . . . 163
10.3.2.2 Организация печати . . . 164
10.3.3 Установка и настройка приложений . . . 165
10.3.3.1 Работа с изображениями . . . 166
10.3.3.2 Броузер . . . 166
10.3.3.3 Почтовый клиент . . . 166
10.3.3.4 Перенос пользовательских настроек . . . 167
10.3.3.5 Пользовательские настройки GNOME . . . 168
10.3.4 Образы экранов: миграция клиента на Linux . . . 169

11 Рекомендации по интеграции **173**
11.1 Подключение к домену Windows . . . 173
11.1.1 Подключение к домену NT4 . . . 173
11.1.2 Подключение к домену Active Directory . . . 174
11.2 Как использовать winbind для подключения локальных пользователей . . . 176
11.2.1 Стандартный winbind . . . 176
11.2.2 Альтернативы winbind . . . 179
11.2.2.1 idmap_rid backend . . . 179
11.2.2.2 idmap_ad backend . . . 179
11.3 Как использовать LDAP для подключения к Active Directory . . . 180
11.4 Pluggable Authentication Modules и домен . . . 182
11.4.1 Аутентификация пользователей с использованием winbind и PAM . . . 183
11.4.1.1 Winbind и PAM в Red Hat Desktop . . . 183
11.4.1.2 Winbind и PAM в Novell Linux Desktop . . . 183
11.4.2 Как производить аутентификацию пользователей через LDAP и PAM . . . 184
11.4.2.1 LDAP и PAM в Red Hat Desktop . . . 184
11.4.2.2 LDAP и PAM в Novell Linux Desktop . . . 186
11.4.3 PAM и домашние каталоги . . . 186
11.4.3.1 Red Hat Desktop . . . 187
11.4.3.2 Novell Linux Desktop . . . 187
11.5 Как подключать файловые ресурсы к Linux-клиенту . . . 187
11.5.1 Подключение ресурсов через smbfs . . . 188
11.5.2 Подключение ресурсов через CIFS . . . 188
11.5.3 Использование smbclient . . . 189
11.6 Автоматическое монтирование домашних каталогов при входе в систему . . . 189
11.6.1 pam_mount на Red Hat Desktop . . . 190

11.6.2 pam_mount в Novell Linux Desktop . . . 192
11.7 Использование сетевых принтеров домена . . . 192
11.7.1 Создание принтера при помощи команды lpadmin . . . 193
11.7.2 Создание принтера через веб-интерфейс CUPS . . . 194

V Приложения **201**

A Словарь Linux для пользователей Windows **203**
A.1 Что все это значит? . . . 203
A.2 Наиболее распространенные термины Linux . . . 204

B Персональные настройки пользователя **217**
B.1 Персональные данные пользователя в Windows . . . 217
B.2 Персональные данные пользователя в Linux . . . 218
B.3 Персональные настройки для рабочего стола под управлением KDE . . . 218
B.4 Персональные настройки для рабочего стола под управлением GNOME . . . 218

C Управление рабочим столом и скрипты **221**
C.1 Языки, ориентированные на создание скриптов . . . 221
C.1.1 Язык сценариев и командный интерпретатор . . . 221
C.1.2 Perl . . . 222
C.1.3 Python . . . 222
C.1.4 Встроенные языки . . . 222
C.1.4.1 Qt Script . . . 222
C.1.4.2 KJSEmbed . . . 223
C.1.5 Kommander . . . 223
C.2 Обмен данными между процессами . . . 224
C.2.1 DCOP . . . 224
C.2.2 Автоматизация рабочего стола с применением DCOP . . . 225

D Перенос приложений **227**
D.1 GTK+ . . . 227
D.2 Qt . . . 227
D.3 REALBasic . . . 227
D.4 wxWidgets . . . 228
D.5 Mono и .NET Framework . . . 228

E Обзор текущего состояния ПО с открытым кодом **229**
E.1 Базовое программное обеспечение . . . 229
E.2 Сообщества разработчиков . . . 229
E.3 Дистрибутивы . . . 230
E.4 Репозитории . . . 231
E.5 Портальные технологии . . . 231
E.5.1 PHP . . . 232
E.5.2 Java . . . 232
E.5.3 Zope . . . 233
E.5.3.1 Python . . . 233
E.5.3.2 Zope . . . 233
E.5.3.3 Plone . . . 233
E.6 ГИС . . . 234
E.6.1 PostGIS . . . 234
E.6.2 MapServer . . . 234
E.6.3 GDAL/OGR . . . 234

F Примеры реализации **235**
F.1 Специализированный дистрибутив . . . 235
F.2 Модель ЦОД . . . 235
F.3 Прототип РОП . . . 236
F.4 Специализированные РОП . . . 238

F.5 Интеграционные порталы . . . 238
F.5.1 Общая архитектура системы . . . 239
F.5.2 Информационный интернет-портал . . . 239
F.5.3 СЭД . . . 241
F.5.4 Сопряжение интеграционного портала и РОП . . . 242
F.6 ГИС . . . 243
F.6.1 Общая архитектура системы . . . 243
F.6.2 Результаты . . . 244
F.7 Автоматизация службы поддержки и сопровождения . . . 244
F.7.1 Развертывание и тестирование ПО . . . 244
F.7.1.1 Клиентская часть . . . 245
F.7.1.2 Серверная часть . . . 245
F.7.2 Портал службы поддержки и сопровождения . . . 245
F.7.2.1 Клиентская часть . . . 246
F.7.2.2 Серверная часть . . . 246

Введение

Торговые марки

Следующие термины являются торговыми марками International Business Machines Corporation в США и/или других странах:

eServer®, Redbooks (logo)™, ibm.com®, AIX®, Domino®, DB2®, DFS™, IntelliStation®, IBM®, Lotus Notes®, Lotus®, Notes®, OS/2®, Redbooks™, S/390®, Sametime®, System i™, System p™, System z™, Tivoli®, WebSphere®, Workplace™, Workplace Client Technology™, Workplace Managed Client™, 1-2-3®.

Следующие термины являются торговыми марками других компаний:

Java, JavaScript, JDK, JVM, NetBeans, Power Management, Sun, Sun Microsystems и все основанные на Java торговые марки являются торговыми марками Sun Microsystems, Inc. в США и/или других странах.

Active Directory, ActiveX, Excel, Expression, Internet Explorer, Microsoft, MSN, Outlook, Visual Basic, Visual C#, Windows Media, Windows NT, Windows Server, Windows Vista, Windows, Win32 и логотип Windows являются торговыми марками Microsoft Corporation в США и/или других странах.

Intel, Itanium, Pentium, Xeon, Intel logo, Intel Inside logo и Intel Centrino logo являются торговыми марками или зарегистрированными торговыми марками Intel Corporation или ее дочерних компаний в США и/или других странах.

UNIX — зарегистрированная торговая марка The Open Group в США и других странах.

Linux — торговая марка Линуса Торвальдса (Linus Torvalds) в США и/или других странах.

AMD и AMD64 logo — торговые марки Advanced Micro Devices Corporation в США и/или других странах.

SUSE и YaST — зарегистрированные торговые марки SUSE AG.

AGFA — торговая марка или зарегистрированная торговая марка Agfa Corporation, Agfa-Gevaert N.V. или Agfa-Gevaert AG в зависимости от юрисдикции.

Nero и Nerovision — зарегистрированные торговые марки Ahead Software AG.

Opera — зарегистрированная торговая марка Opera AS.

Novell, iFolder, Mono, Nterprise, Red Carpet, Ximian, Novell Evolution и ZENworks — торговые марки или зарегистрированные торговые марки Novell в США и/или других странах.

Red Hat — торговая марка Red Hat Corporation в США и/или других странах.

Tarantella — торговая марка или зарегистрированная торговая марка Tarantella Corporation в Соединенных Штатах Америки и в других странах.

Debian — зарегистрированная торговая марка Software in the Public Interest, Inc.

OpenLDAP — зарегистрированная торговая марка OpenLDAP Foundation, Inc.

Mandrake — зарегистрированная торговая марка Mandrakesoft S. A. и Mandrakesoft Corporation.

REALbasic — зарегистрированная торговая марка Real Software Corporation.

VMWare — зарегистрированная торговая марка VMWare Corporation.

KDE, K Desktop Environment и Konqueror — зарегистрированные торговые марки KDE e.V.

GNOME — торговая марка GNOME Foundation.

Apple и Macintosh — торговые марки Apple Computer Corporation в США и/или других странах.

Macromedia и Macromedia Flash — торговые марки Macromedia Corporation в США и других странах.

Mozilla, Mozilla Firefox и Mozilla Thunderbird — торговая марка или зарегистрированная торговая марка The Mozilla Organization в США и/или других странах.

Adaptec и Easy CD Creator — торговые марки или зарегистрированные торговые марки Adaptec Corporation в США и/или других странах.

Adobe, Acrobat, Photoshop, FrameMaker и Adobe Reader — или зарегистрированные торговые марки, или торговые марки Adobe Systems Incorporated в США и/или других странах.

Citrix и MetaFrame — или зарегистрированные торговые марки, или торговые марки Citrix Systems Incorporated в США и/или других странах.

NoMachine и NoMachine NX — торговые марки или зарегистрированные торговые марки Medialogic S.p.A Corporation Италии.

SAP — торговая марка или зарегистрированная торговая марка SAP AG в Германии и некоторых других странах.

RealPlayer — зарегистрированная торговая марка RealNetworks Incorporated в США и/или других странах.

WinZip — зарегистрированная торговая марка Winzip Computing Incorporated.

Yahoo! — зарегистрированная торговая марка Yahoo! Incorporated.

Big Brother — торговая марка Quest Software Corporation.

Nagios — зарегистрированная торговая марка Ethan Galstad.

Netscape — зарегистрированная торговая марка Netscape Communications Incorporated в США и/или других странах.

Jasc и Jasc Software — зарегистрированные торговые марки Jasc Software Corporation.

CUPS и Common UNIX Printing System — торговые марки Easy Software Products Co.

SET, SET Secure Electronic Transaction и SET Logo — торговые марки, принадлежащие SET Secure Electronic Transaction LLC.

Названия других компаний, продуктов или сервисов могут являться чьими-то торговыми марками или сервисными марками.

Об авторах книги IBM Redbook

Команда авторов IBM Redbook

Вторая версия книги IBM Redbook была создана командой специалистов со всего мира, работающих в International Technical Support Organization, Austin Center.

Крис Алмонд (Chris Almond) — руководитель проекта и IT Architect в ITSO, Остин, штат Техас. Он имеет 15-летний опыт работы в ИТ-индустрии, из них последние 5 лет в IBM. Имеет опыт работы с системами UNIX/Linux, сетевыми окружениями, Lotus® Domino®-based content management solutions и разработки портальных решений на базе WebSphere.

Джероен ван Хоф (Jeroen van Hoof) — старший ИТ-специалист в IBM Global Services в Нидерландах. Он имеет более чем 15-летний опыт работы с операционными системами UNIX и более чем 10-летний опыт работы с Linux. До объединения IBM с PwC Consulting он работал в сфере высокопроизводительных вычислений в Dutch National Supercomputing Centre в качестве эксперта по параллельными вычислениям. Он имеет опыт работы с UNIX/Linux, Windows, системами виртуализации, сетями TCP/IP (routing, switching), параллельным вычислениям (MPI, PVM), механизмами защиты firewall, опыт программирования (Fortran, C, Pascal, Perl), SAP Basis consultancy и использования Web-технологий (HTTP, HTML, CGI). Имеет статусы Red Hat Certified Engineer, VMWare Certified Professional и SAP Certified Technology Consultant. Имеет степень доктора философии в области теоретической физики в университете Неймегена (University of Nijmegen).

Ник Лассонде (Nick Lassonde) — Chief Software Architect в Versora, где руководит разработками Progression Desktop, передового средства автоматизации для переноса персональных данных между Windows и Linux окружениями. До Versora Ник был старшим архитектором в Miramar Systems, где занимался сначала разработкой средств для переноса критических данных и настроек между вычислительными операционными системами и проектировал и осуществлял процедуры автоматизированного тестирования. До Miramar Ник работал в качестве архитектора программного обеспечения в Cadence Design Systems, где занимался вопросами взаимодействия между рабочими станциями под управлением Microsoft и Linux/UNIX®.

Бен Ли (Ben Li) — вице-президент Innovation and Outreach at Userful Corporation, где руководит разработкой интегрированного программного обеспечения и технического обеспечения для multi-station Linux systems. Имеет технический опыт управления в сфере обеспечения безопасности и системного администрирования в системах Linux, Windows и OS X в высших учебных заведениях, разработке и детализации систем управления контентом и систем workflow, а также он-лайн и cross-media публикации. Бен также является исследователем в «The Center for Innovation Studies», базирующейся в Калгари исследовательской и адвокатской группе по инновациям и исследованиям в области коммерциализации комьюнити, и имеет две ученых степени в университете Калгари. В качестве консультанта до присоединения к Userful Бен разрабатывал «helpdesk notification system» для многонациональной корпорации бизнес-сервисов, а также построенную с использованием базы данных систему для отслеживания связей с общественностью для ведущей Канадской энергетической компании.

Курт Тейлор (Kurt Taylor) — старший разработчик программного обеспечения в IBM Linux Technology Center, расположенном в Остине, штат Техас. В настоящее время Курт является руководителем технической группы в Linux System Management Team. Он проработал в IBM 10 лет и имеет более чем 17-летний опыт промышленного проектирования, создания и разработки распределенных сетей и продуктов для управления системами. В настоящее время работает над проектом OpenPegasus и является членом рабочей группы OSDL Desktop Linux. Курт закончил университет северного Техаса и имеет степень бакалавра компьютерных наук.

Благодарности

Команда второй версии IBM Redbook выражает признательность и благодарность команде авторов первой версии книги: Крису Олмонду (Chris Almond), Арту Кэннону (Art Cannon), Джероену ван Хофу (Jeroen van Hoof), Кристиану Пачу (Christian Patsch), Секкару Ваддади (Sekkar Vaddadi), Оливеру Марку (Oliver Mark) и Томасу Шваллеру (Thomas Schwaller). Их усилия при написании первой версии помогли в создании оригинальной концепции этой книги и определили ее структуру. Значительная часть содержимого первой версии присутствует и во второй версии книги.

Кроме того, команда хотела бы выразить благодарность за существенную поддержку этого проекта следующим людям:

- Грегу Келлеру (Greg Kelleher), Senior Program Manager Worldwide Linux Desktop Strategy and Market Development: за спонсирование проекта и помощь в руководстве разработкой проекта содержимого и концепции этой книги.
- Стивену Хохстетлеру (Stephen Hochstettler), ITSO Project and Linux Team Leader: за его видение этого проекта и поддержку.
- Джону Бергланду (John Bergland), ITSO Project Leader, Lotus и WebSphere Portal, за его вклад.
- Бернарду Голдену (Bernard Golden), за сотрудничество.
- Гарри Андерсену (Gerry Anderson), Джиму Землину (Jim Zemlin) и Дирку Хонделу (Dirk Hohndel), Intel® Inc.
- Майку Шиффи (Mike Sheffey), Джону Уолкеру (Jon Walker), Versora Inc., и Тимоти Гриффину (Timothy Griffin), Userful Inc.
- Джерри Риверосу (Gerry Riveros), Джонатану Блэндфорду (Jonathan Blandford), Крису Близзарду (Chris Blizzard), Дэну Уолшу (Dan Walsh), Сету Никеллу (Seth Nickell) и Мартину Даффи (Máirín Duffy), Red Hat, Inc.
- Профессионалам IBM ITSO: Эрике Вазевски (Erica Wazewski) (юридическое сопровождение) и Лесли Пархам (Leslie Parham) за помощь в подготовке книги к публикации.

Об авторах русской части руководства

В подготовке и составлении материалов, представленных в этом руководстве, принимали участие:

- руководство проектом — Олег Садов;
- перевод — Екатерина Смирнова, Надежда Черниговская;
- техническое редактирование, подготовка руководства в формате DocBook XML — Елена Корытько, Татьяна Нагорская;
- техническое сопровождение, подготовка оригинал-макета — Владимир Титов.

Оказали неоценимую помощь в подготовке материалов, вошедших в данное руководство, и приняли участие в создании прототипов и их апробации сотрудники администрации Новгородской области и города Великого Новгорода. Особенно хотелось бы отметить участие Новгородского Информационно-Аналитического Центра.

Сердечную благодарность хотелось бы выразить Дмитрию Дмитриеву, Сергею Егорову, Михаилу Компанийцу, Станиславу Латышко, Алексею Лобашеву, Василию Пантюхину, Даниилу Плитману, Артему Третьяченко и Дмитрию Хорошавину за участие в подготовке материалов для создания данного руководства.

Основные термины и определения

Информация — сведения о лицах, предметах, фактах, событиях, явлениях и процессах независимо от формы их представления.

Информатизация — организационный социально-экономический и научно-технический процесс создания оптимальных условий для удовлетворения информационных потребностей и реализации прав граждан, органов государственной власти, органов местного самоуправления, организаций, общественных объединений на основе формирования и использования информационных ресурсов.

Документированная информация (документ) — зафиксированная на материальном носителе информация с реквизитами, позволяющими ее идентифицировать.

Информационные процессы — процессы сбора, обработки, накопления, хранения, поиска и распространения информации.

Информационная система — организационно упорядоченная совокупность документов (массивов документов) и информационных технологий, в том числе с использованием средств вычислительной техники и связи, реализующих информационные процессы.

Информационные ресурсы — отдельные документы и отдельные массивы документов, документы и массивы документов в информационных системах (библиотеках, архивах, фондах, банках данных, других информационных системах).

Информация о гражданах (персональные данные) — сведения о фактах, событиях и обстоятельствах жизни гражданина, позволяющие идентифицировать его личность.

Конфиденциальная информация — документированная информация, доступ к которой ограничивается в соответствии с законодательством Российской Федерации.

Средства обеспечения автоматизированных информационных систем и их технологий — программные, технические, лингвистические, правовые, организационные средства (программы для электронных вычислительных машин; средства вычислительной техники и связи; словари, тезаурусы и классификаторы; инструкции и методики; положения, уставы, должностные инструкции; схемы и их описания, другая эксплуатационная и сопроводительная документация), используемые или создаваемые при проектировании информационных систем и обеспечивающие их эксплуатацию.

Собственник информационных ресурсов, информационных систем, технологий и средств их обеспечения — субъект, в полном объеме реализующий полномочия владения, пользования, распоряжения указанными объектами.

Владелец информационных ресурсов, информационных систем, технологий и средств их обеспечения — субъект, осуществляющий владение и пользование указанными объектами и реализующий полномочия распоряжения в пределах, установленных Законом.

Пользователь (потребитель) информации — субъект, обращающийся к информационной системе или посреднику за получением необходимой ему информации и пользующийся ею.

Список сокращений

ПО — Программное обеспечение

ИС — Информационная система

ИКТ — Информационно-коммуникационные технологии

СУБД — Система управления базами данных

ОС — Операционная система

БД — База данных

АРМ — Автоматизированное рабочее место

БПТИ — Базовая программно-технологическая инфраструктура

ГИС — Геоинформационная система

ИКТ — Информационно-коммуникационные технологии

ПК — Персональный компьютер

РОП — Рабочее окружение пользователя

СЭД — Система электронного документооборота

ФЦП — Федеральная Целевая Программа «Электронная Россия»

АРМ — Автоматизированное рабочее место

Информационные и телекоммуникационные технологии в России

В современном стремительно развивающемся мире функционирование и развитие государства немыслимо без применения современных средств и технологий управления, коммуникации и связи. Как показывают многочисленные исследования, увеличение объемов информации, циркулирующей в сфере управления, принимает взрывной характер. В этих условиях управление без применения современных средств и способов обработки, передачи и представления информации не может быть эффективным.

О важности современного подхода к процессу управления говорится в федеральной целевой программе (ФЦП) «Электронная Россия»:

> Развитие и широкое применение информационных и коммуникационных технологий (далее — ИКТ) является глобальной тенденцией мирового развития и научно-технической революции последних десятилетий. Применение ИКТ имеет решающее значение для повышения конкурентоспособности экономики, расширения возможностей ее интеграции в мировую систему хозяйства, повышения эффективности государственного управления и местного самоуправления.
>
> Важным результатом распространения ИКТ и проникновения их во все сферы общественной жизни является создание правовых, организационных и технологических условий для развития демократии за счет реального обеспечения прав граждан на свободный поиск, получение, передачу, производство и распространение информации.
>
> —ФЦП «Электронная Россия»

Исторически сложилось так, что широкое применение ИКТ в России в области государственного управления началось со значительным отставанием по сравнению с другими государствами. В настоящее время интенсивность применения современных ИКТ в России по ряду причин недостаточно высока:

> Несмотря на высокие темпы развития ИКТ в последнее десятилетие, Россия не смогла сократить отставание от промышленно развитых стран в уровне информатизации экономики и общества. Отчасти такое положение вызвано общеэкономическими причинами (длительный кризис в экономике, низкий уровень материального благосостояния большинства населения). Вместе с тем недостаточное развитие ИКТ в России усугубляется целым рядом факторов, создающих препятствия для широкого внедрения и эффективного использования ИКТ в экономике, развития производства в сфере ИКТ. К числу таких негативных факторов относятся:
>
> - несовершенная нормативная правовая база, разрабатывавшаяся без учета возможностей современных ИКТ;
> - недостаточное развитие ИКТ в области государственного управления, неготовность органов государственной власти к применению эффективных технологий управления и организации взаимодействия с гражданами и хозяйствующими субъектами;
> - отсутствие целостной информационной инфраструктуры и эффективной информационной поддержки рынков товаров и услуг, в том числе в сфере электронной торговли;
> - недостаточный уровень подготовки кадров в области создания и использования ИКТ;
> - барьеры, возникающие из-за недостатков в регулировании экономической деятельности при выходе российских предприятий и других организаций сферы ИКТ на российский и мировой рынки;
> - высокий уровень монополизации сетей связи, создающий барьеры на пути их использования и приводящий к перекосам в тарифной политике.
>
> Процессы информатизации уже активно идут на всех уровнях, многие мероприятия, направленные на развитие ИКТ, реализуются или планируются к реализации в рамках федеральных, региональных и ведомственных программ.

Для придания ускорения процессу широкого внедрения ИКТ в сфере управления государством была принята Федеральная целевая программа «Электронная Россия (2002-2010 годы)» (далее — Программа), которая «должна обеспечить формирование нормативной правовой базы в сфере ИКТ, развитие информационной и телекоммуникационной инфраструктуры, сформировать условия для подключения к открытым информационным системам (в том числе посредством сети Интернет), а также обеспечить эффективное взаимодействие органов государственной власти и органов местного самоуправления с гражданами и хозяйствующими субъектами на

основе широкого внедрения ИКТ. В процессе выполнения Программы будут определяться общие направления развития ИКТ (основные принципы, стандарты и типовые решения по реализации различных проектов) как одного из основных направлений социально-экономического развития страны.»

Вопросы использования информационных технологий в деятельности федеральных органов государственной власти находятся под пристальным вниманием правительства РФ. Так, «Концепция использования информационных технологий в деятельности органов государственной власти до 2010 года», одобренная распоряжением Правительства Российской Федерации от 27 сентября 2004 г. N 1244-р, гласит:

> Основной целью использования информационных технологий в деятельности федеральных органов государственной власти является повышение эффективности механизмов государственного управления на основе создания общей информационно-технологической инфраструктуры, включающей государственные информационные системы и ресурсы, а также средства, обеспечивающие их функционирование, взаимодействие между собой, населением и организациями в рамках предоставления государственных услуг.
>
> Государственная политика в сфере использования информационных технологий в государственном управлении призвана обеспечить координацию деятельности федеральных органов государственной власти по созданию государственных информационных систем и повышение эффективности бюджетных расходов в этой сфере.
>
> В результате реализации государственной политики ожидается формирование эффективной системы предоставления государственных услуг на основе использования информационных технологий (далее — «электронное правительство»).

Использование информационных технологий в государственном управлении и развитие информационно-технологической инфраструктуры федеральных органов государственной власти осуществляются в соответствии с планом создания «электронного правительства».

Текущее состояние региональных администраций и муниципальных образований в области автоматизации собственной деятельности характеризуется несколькими факторами. С одной стороны, динамично развивающаяся общественная и хозяйственная жизнь требует более широкой автоматизации управленческой деятельности, что приводит к бурному развитию внедряемых в области и муниципальных образованиях решений в области информационных технологий. С другой стороны, активно развивающийся рынок сетевых и компьютерных технологий требует коренной модернизации внедренных информационно-технических решений каждые 3-5 лет для предотвращения их морального устаревания.

Применительно к данному документу информатизация — это процесс создания благоприятных условий для наиболее полного удовлетворения информационных потребностей властей и населения.

Цели этого процесса:

1. обеспечение актуальной информацией как работников органов управления различных уровней для эффективного выполнения возложенных на них функций, так и населения города и области с соблюдением требований действующего законодательства;

2. контроль состояния, научно-обоснованное прогнозирование и комплексное планирование социально-экономического развития;

3. реализация законодательных и нормативно-правовых мер, вырабатываемых федеральными, региональными и местными органами власти и управления, а также инициативная подготовка предложений указанным органам по совершенствованию законодательной и нормативной базы;

4. оперативная подготовка управленческих решений и мер по их реализации со стороны властей и прозрачность власти для населения.

В связи с этим крайне важным является широкомасштабное внедрение информационных технологий во все свои основные процессы управления, причем в ФЦП «Электронная Россия» отдельно отмечается необходимость преимущественного использования алгоритмов и программ для ЭВМ, тексты которых открыты и общедоступны в деятельности органов государственной власти и органов местного самоуправления. Использование программного обеспечения разработанного в рамках проектов по созданию ПО с открытым кодом (Open Source) позволяет эффективно и в сжатые сроки решить задачу коренной модернизации ИКТ-инфраструктуры органов власти.

Предпосылки для использования ПО с открытым кодом

Программное обеспечение с открытым кодом (Open Source) предполагает поставку исходных кодов программного продукта потребителю данного продукта. Модель свободного распространения (Free Software) таких продуктов предполагает возможность их получения и использования без осуществления каких-либо выплат любым желающим. Обычно и сами проекты по созданию таких продуктов осуществляются сообществом разработчиков на добровольной основе, хотя нередки случаи спонсирования таких проектов или отдельных разработчиков государственными структурами или частными компаниями. Продвижением продукта на рынке и созданием сервисов вокруг него занимаются коммерческие структуры, в том числе и такие крупные игроки на ИТ-рынке как IBM, Novell, Red Hat и т.д.

Все более широкое распространение такое ПО получает в государственных структурах различных стран. Около половины местных органов власти в Европе используют open source программы и в тоже время треть даже не знает что они используют open source. По результатам опроса Маастрихтского Университета (Maastricht University) в 12 странах 49% респондентов используют Free/Libre/Open Source Software (FLOSS).

Объединенный отчет университета и Международного Института Экономики (International Institute of Infonomics), показал что важным при выборе являлось: производительность для 83%, безопасность для 75%, и низкие лицензионные отчисления для 71%.

Не меньшее значение внедрение свободно-распространяемого ПО с открытыми кодами имеет и для России. Эффекты от такого внедрения будут носить общегосударственный характер.

Технологические

Развитие Интернет сократило время на информационные коммуникации между людьми и обеспечило быстрый доступ к большим объемам информации. Этот факт и существенные инвестиции мировых лидеров отрасли информационно-коммуникационных технологий (ИКТ) обеспечили появление качественного ПО с открытыми кодами, которое можно использовать и модернизировать совершенно легально без каких-либо лицензионных платежей.

Правовые

В настоящее время на территории России большинство пользователей компьютеров, как в государственных учреждениях, так и в частном/индивидуальном секторе, пользуется нелицензионным ПО.

Экономические

В настоящее время вырученные денежные средства за ПО «уходят» за рубеж, стимулируя развитие отрасли ИКТ там, а не в России.

Исторические

Вследствие феномена свободного программного обеспечения — возможности воспользоваться на легальной основе через Интернет разработанным квалифицированными специалистами ПО — Россия именно сейчас получает возможность быстро, с наименьшими расходами создать собственную отрасль по производству ПО.

Политические

Происходит интеграция России в мировое сообщество, при этом свободное ПО может обеспечить решение вопроса: как одновременно соответствовать общемировым информационным стандартам и обеспечивать информационную безопасность страны.

Социальные

Широкое использование открытых программных продуктов позволит осуществить вовлечение в процесс популяризации информационных технологий школьников и студентов, которые через какое-то время станут определять дальнейшую политику страны в области информатизации и использования информационных технологий, как это происходило в развитых странах мира, когда основным каналом продвижения ИТ были университеты, колледжи, школы; малоимущие слои населения, что позволит им, получив дополнительные знания, применить их в быстрорастущем секторе рынка информационных услуг.

Эффекты, получаемые при использовании свободного ПО с открытым кодом

Экономический

- достижение глобальной экономии государственных (бюджетных) средств на поддержание БПО в государственных учреждениях — за счёт снижения затрат на приобретение лицензий у западных производителей;
- высвобождение средств на развитие собственной ИТ индустрии, что обеспечит конкурентоспособность отечественных разработок;
- смещение акцентов в народном хозяйстве России с развития сырьевых отраслей на развитие наукоемких отраслей;
- качественное развитие сферы информационных услуг;
- повышение прозрачности деятельности предприятий госсектора;
- в среднесрочной перспективе — увеличение объёма налогов, собираемых не только в отечественной ИТ отрасли, но и в целом в народном хозяйстве;
- развитие данного направления ИТ отрасли способствует образованию необходимого количества квалифицированных специалистов, обеспечивающих внедрение современных информационных технологий на предприятиях;
- кардинальное снижение уровня пиратства позволит России стать полноценным членом мирового сообщества, в частности будет способствовать вступлению в ВТО.
- образование новых рабочих мест;
- наращивание интеллектуального потенциала в России — воспитание большого количества высококвалифицированных программистов;
- решение проблемы занятости и проведения досуга молодежи.

Укрепление национальной безопасности

- обеспечение безопасности государства — полный, реальный контроль над качеством используемого в государственных учреждениях, в том числе силовых структурах, программного обеспечения;
- повышение экономической безопасности в силу отсутствия зависимости от иностранных производителей и поставщиков программного обеспечения.

Технологический

- возможность быстрой технологической адаптации и модернизации БПО;
- более эффективная работа БПО в государственных учреждениях благодаря техническим особенностям БПО: уникальной масштабируемости, отсутствию компьютерных вирусов, широкому диапазону использования БПО — от встраиваемых портативных систем до суперкомпьютеров;

- предоставление унифицированных решений на различных уровнях (предприятие, регион, федерация) с использованием всех преимуществ, даваемых современными информационными технологиями;
- возможность устойчивой работы и обеспечение удаленного доступа к объединенным информационным государственным ресурсам, в том числе в рамках программы «Электронная Россия»;
- возможность влияния на мировой ИТ рынок.

Развитие науки и образования:

- активное вовлечение научных и учебных заведений всей страны в процесс разработок, что позволяет реализовать масштабные общероссийские проекты;
- повышение качества образования, как в средних, так и в высших и специальных учебных заведениях

Широкое внедрение свободно-распространяемого ПО с открытыми кодами будет способствовать быстрому росту информационной и экономической независимости и, следовательно, безопасности России.

Экономическое обоснование использования разработок с открытым кодом

Использование свободного ПО — наиболее быстрый и эффективный путь внедрения в общемировой рынок ПО. Индустрия развития информационных технологий — на данный момент наиболее быстро развивающийся сектор мировой экономики. Состояние рынка в этой области меняется очень быстро, и у России есть реальный шанс занять в нем одну из лидирующих позиций, особенно в сфере разработки ПО. Этому способствует наличие у нас в стране большого числа высококвалифицированных специалистов в данной области.

В настоящий момент на мировом рынке ПО сложилась уникальная ситуация — США более не является монополистом в области передовых программных разработок. Все больше программных проектов, как коммерческих, так и свободных, создается и развивается за пределами США. Linux был рожден в Финляндии, GNOME в Бразилии, KDE в Германии. Большинство разработок делаются с привлечением специалистов из самых разных стран мира. В России огромный потенциал разработчиков. В большинстве международных проектов по разработке свободного ПО участвуют российские разработчики.

Главные моменты при выборе ПО для государственных органов: во–первых, обеспечение информационной безопасности страны, во-вторых, деньги должны оставаться в России. Вместо обязательных платежей за лицензии государство сможет вкладывать средства в сферу образования, в создание рабочих мест для отечественных специалистов. При государственном финансировании разработок следование открытым стандартам (например, POSIX) и открытость кода должны быть обязательными требованиями.

Известны две схемы развития индустрии отечественного ПО:

1. оффшорное программирование;
2. разработка собственных продуктов и распространение их на общемировом рынке товаров и услуг.

До сих пор развитие индустрии ПО в России в большинстве случаев идет по первому пути, когда результаты разработок оказываются собственностью зарубежных компаний и через какое-то время поступают на российский рынок как импортируемый продукт. Второй путь гораздо перспективнее, но требует финансирования фундаментальной науки (там отрабатываются новые идеи) и инвестиций (в том числе и государственных) в разработку конечного продукта.

Крайне важным для развития индустрии является правильный выбор политики государства в области поддержки открытых стандартов. Уже сейчас в стране существует ряд достаточно успешно работающих фирм-разработчиков программного обеспечения, создающих собственные программные продукты. К сожалению, большинство из них в качестве базового программного уровня используют закрытые коммерческие системы иностранного производства (например, Microsoft Windows). Это не способствует повышению экономической безопасности страны, ставя ее в прямую зависимость от интересов фирм производителей и государственной политики тех стран, откуда производится экспорт таких систем.

Возможные пути развития новых технологий — корпоративные разработки, государственное финансирование (гранты, вузы, специализированные институты), деятельность независимых разработчиков. Мировой опыт показывает: большинство инновационных технологических разработок рождается в вузах, зачастую как побочный результат фундаментальных исследований. Существенную долю в разработку нового ПО вносят студенты,

занимаясь этим в свободное время и повышая таким образом свой статус. Коммерческие структуры редко тратят деньги на действительно инновационные исследования, обычно они используют уже готовые результаты научных исследований.

Развитие ПО очень похоже на научный процесс. Общая открытость его только ускоряет. Лучшая черта открытого ПО — расширяемость. Типичная ситуация — 80% требуемой функциональности уже есть в каких-то продуктах и требуется сравнительно небольшая их доработка для решения новых задач.

Открытое ПО дает конкурентные преимущества даже в производстве коммерческого ПО. Большинство ведущих ИТ корпораций всё шире используют открытые программные продукты, разработанные мировым сообществом разработчиков, или же открывают исходные тексты своих продуктов. Покупатель больше не хочет платить за отдельные компоненты, предпочитая целостные решения и сервис. При этом достигается большая независимость от производителей и сохраняется полный контроль со стороны покупателя над приобретаемыми продуктами.

Вопросы безопасности

Использование ПО с открытыми кодами может дать гарантию безопасности при внедрении информационных технологий.

Только при таком подходе возможны:

полный анализ содержимого исходных кодов;

- оперативное приведение их в полное соответствие с государственными правовыми и нормативными требованиями;
- контроль за ходом технологического процесса на всех этапах от разработки до окончательного внедрения.

При таком подходе принципиально невозможно использование со стороны поставщика или изготовителя ПО так называемых «программных закладок», способных производить неподконтрольную передачу внутренней информации или преднамеренно нарушать работу вычислительных комплексов.

Открытое ПО предоставляет России реальную независимость как от коммерческой политики фирм производителей, так и от ограничений, накладываемых на государственном уровне странами экспортерами программных продуктов.

Правовые аспекты

Кардинально решить проблему пиратства можно лишь при условии широкого внедрения свободно распространяемого ПО с открытым кодом. Это не требует внешних лицензионных выплат и позволяет продолжать пользоваться привычным набором ПО абсолютно легально.

Обычно Open Source проекты распространяются по одной из лицензий, отвечающей требованиям Open Source Initiative (http://www.opensource.org), хотя некоторые публикуются и под двойными лицензиями, допускающими их коммерческое использование в закрытых продуктах.

Существуют различные варианты свободных лицензий и различные варианты их использования. Наибольшее распространение получили два типа свободных лицензий. Условно их можно охарактеризовать следующим образом:

1. Универсальная общественная лицензия GNU (GNU General Public License, сокращенно GPL). Разрешает свободное использование исходных кодов программных продуктов, попадающих под ее область действия (в том числе и в коммерческих целях), и внесение в них любых изменений. В случае использования их в своих разработках разработчик обязуется в дальнейшем предоставлять свои исходные коды по первому требованию.
2. Программная лицензия университета Беркли (Berkeley Software Distribution, сокращенно BSD). Так же предоставляет право неограниченного использования в сторонних разработках, но, в отличие от GPL, позволяет в дальнейшем сделать продукт закрытым.

Другие свободные лицензии (Artistic, Apache, LGPL, MIT/X11, ZPL и др.) в той или иной степени дополняют изложенные выше требования одной или другой лицензии.

Использование в качестве базы открытого программного обеспечения никоим образом не ущемляет интересы производителей программных продуктов. Даже при использовании программных продуктов под лицензией GPL производитель не обязательно должен открывать свой код (например, в том случае, когда используются разделяемые библиотеки или создаются подгружаемые модули ядра ОС). Использование таких продуктов во внутренних проектах организаций/фирм, так же не накладывает на них обязательств по необходимости открытия исходных кодов.

Перспективы

В мире в настоящее время определились три глобальных процесса:

- демократические преобразования, причем, не только как способ осуществления власти, но и как участие граждан в анализе различной информации, оценке событий и выражение этой оценки;
- распространение экономического либерализма при понимании того, что бесконтрольность глобальной конкуренции может способствовать социальной и политической неустойчивости;
- неизбежная революция управления: классические государства как системы власти не могут удерживать тот объем исключительной власти, каким обладали недавно (третий процесс во многом является следствием первых двух).

Распространилось явно сформулированное понимание того, что государственные органы должны непосредственно работать на граждан и страну, что получило название «услуги государства» гражданам, частному бизнесу общественным организациям и стране в целом, а потребители услуг государства стали называться клиентами или пользователями.

Вопросы улучшения системы оказания государственных услуг занимают важное место в политике многих государств. В последнее время этим вопросам все большее внимание уделяется и в России. В этих условиях реформирование органов государственной власти потребовало резкого увеличения применения ИКТ и качественного изменения ее роли. Причины этого в следующем:

- необходимость предоставления доступа граждан к открытой информации государственных органов и организаций;
- необходимость сокращения затрат труда чиновников на предоставление государственных услуг гражданам, бизнесу и другим подразделениям органов власти;
- потребность сокращения потерь времени при обращении в государственные органы;
- организация горизонтальных связей между чиновниками одного или различных ведомств для оперативного решения комплексных задач в режиме сотрудничества и др.

В нашей стране в настоящее время созданы условия для построения современной ИТ-инфраструктуры. Реализация этих условий возможна лишь на основе комплексного подхода, с использованием современных технических и технологических решений. Данное руководство содержит методические рекомендации по анализу ИТ-инфраструктуры региональной администрации и определению перспективных направлений ее модернизации, на основе программного обеспечения с открытым кодом.

Такое ПО уже завоевало достаточную популярность в среде ИТ-профессионалов и занимает достойное место в качестве базовой платформы при построении серверных комплексов. Так, согласно данным, приведенным на сайте компании Netcraft http://netcraft.com, более 60% веб-серверов в Интернете работают под управлением систем построенных на базе Open Source продуктов. В отличие от серверов, на рабочих компьютерах пользователей, данные системы еще достаточно редки, в первую очередь в силу малого количества литературы по этому вопросу и, как следствие, слабой осведомленности простых пользователей и ИТ-специалистов о возможностях систем, построенных на базе ОС Linux.

О данном руководстве

Данное руководство не следует рассматривать как исчерпывающее описание всех программных продуктов и технологий доступных в мире Open Source. Только на одном из хостингов для Open Source проектов, SourceForge (http://www.sourceforge.net), поддерживается более сотни тысячи таких проектов. И надо заметить, что ландшафт этого мира постоянно меняется, поскольку движение Open Source носит крайне динамичный характер и очень быстро откликается на все новейшие идеи и разработки. Главное, на чем нам хотелось сосредоточиться в этом руководстве, это на самой философии и методологии использования Open Source продуктов. Различия между традиционных подходах при реализации ИТ-проектов и использованием Open Source отражены в Табл. 1.

Таблица 1: Возможные подходы при разработке и внедрении программных решений

Традиционная модель	Open Source
Закупка лицензий на закрытые программные продукты	Поиск и анализ открытых программных решений
Разработка собственных программных решений с нуля	Доработка существующих программных продуктов и их комбинаций

Оба этих подхода не противоречат, а скорее взаимно дополняют друг друга. Главное изменение в современной парадигме информатизации заключается в том, что наряду с традиционными методами проведения работ в этой сфере требуется включение этапа активного сбора данных о существующих открытых программных разработках и проведение анализа возможности их использования в текущих и перспективных проектах. Краткий обзор ресурсов, связанных с Open Source проектами, можно найти в Прил. E.

Поскольку правильный выбор клиентской платформы и составление четкого плана миграции являются крайне важными аспектами при проведении комплексной модернизации ИТ-инфраструктуры, особенно в органах государственного управления и местного самоуправления, мы уделили довольно много внимания этим вопросам и включили в данное руководство довольно большое количество материалов, представленных в издании IBM RedBook *Linux Client Migration Cookbook, A Practical Planning and Implementation Guide for Migrating to Desktop Linux*:

http://www.redbooks.ibm.com/abstracts/sg246380.html

Кроме того, в руководстве представлены материалы, полученные компанией «ОАО Линукс Инк», см.:

http://www.linux-ink.ru

в ходе проведения работ по созданию концепции модернизации ИТ-инфраструктуры органов управления Новгородской области (проект выполнялся при поддержке Министерства Экономического Развития и Торговли), а так же ряда других проектов, выполнявшихся этой компанией. Примеры применения Open Source решений в реальных проектах можно найти в Прил. F

Данное руководство подготовлено в программной среде Scientific Linux 4x в формате DocBook XML, с использованием программных пакетов: docbook-utils, db2latex, Emacs, PSGML, LaTeX2e, Openoffice.org, GIMP, Dia.

Часть I

Модернизация ИКТ-инфраструктуры

Глава 1

Текущее состояние информационных систем

1.1 Обеспечение доступа граждан к государственным информационным ресурсам

Изменения экономической ситуации в существенной степени затрагивают социальные и гуманитарные аспекты жизни общества, выдвигая на первый план вопросы социальной защиты населения, связанные с угрозой роста безработицы, непрерывным ростом стоимости жизни, а также значительными изменениями в организации системы образования и здравоохранения. Регулирование перечисленных процессов требует массового использования информационных технологий, обеспечивающих социальный и экономический мониторинг.

В этих условиях важно оперативное информационно-аналитическое обслуживание органов государственной власти и управления. Одной из основных проблем в процессе информатизации их деятельности является обеспечение взаимодействия и интеграции существующих межведомственных и ведомственных автоматизированных информационных систем и их информационных ресурсов.

Проводимые в настоящее время работы по информатизации в рамках отдельных отраслей и ведомств слабо скоординированы между собой и не согласованы с задачами информатизации региона. Это существенно снижает общий эффект производимых затрат и оставляет в стороне насущные интересы региона в области информатизации органов власти и управления, социальной сферы, экологии, обслуживания населения и т.п.

1.2 Структура информационного обмена

С учетом специфики организации работ в подразделениях органов государственного управления информационные потоки организуются по двухуровневой схеме: областной уровень — районный уровень.

В общем случае информационный обмен в системе можно рассматривать как совокупность следующих потоков информации.

- Потоки с детерминированными маршрутами:
  - восходящий — обеспечивающий сбор первичной информации, отчетности и запросов на получение информационно-справочных материалов (пример: район-область);
  - нисходящий — обеспечивающий доведение до исполнителей директивных материалов, управленческих решений, информационно-справочных материалов, необходимых для функционирования органов регионального управления определенного уровня.
- Потоки с непрогнозируемыми маршрутами:
  - горизонтальный поток внутри свободно организуемых отраслевых подсистем управления;
  - произвольно-ориентированный поток прямой переписки субъектов системы управления.

Горизонтальный поток координирует доминирующий в отраслевой подсистеме орган (департамент, предприятие, информационное агентство и т.д.).

Часть нисходящего потока, связанного с доведением директивных материалов и управленческих решений, координирует орган, инициирующий эти материалы.

Другая часть нисходящего потока, связанная с поставкой информационных материалов, а также весь восходящий поток на уровне областного центра координируется аналитическими подразделениям органов управления.

Анализ документооборота в органах управления области показывает, что наиболее интенсивными являются информационные потоки с непрогнозируемыми маршрутами. Интенсивность детерминированных потоков типа область-район примерно на порядок ниже.

В структуре информационного обмена можно выделить ряд потоков, преобладающих на всех уровнях обмена. Очевидно, что источники-приемники таких информационных потоков должны быть первоочередными абонентами региональной информационно-телекоммуникационной сети.

Анализ объемов информационного обмена показывает, что в детерминированных потоках преобладают потоки с объемом до 50 док/год; в потоках с непрогнозируемыми маршрутами распределение по объемам более равномерно и преобладающими являются потоки с объемами до 500 док/год.

Помимо учтенного документооборота информационный обмен осуществляется оперативным путем средствами телекоммуникаций в виде информационных контактов в вычислительных сетях, на рабочих совещаниях. Анализ структуры и объемов такого обмена затруднен, но общепринятой является оценка соотношения данного вида информационного обмена к документообороту, как 10:1.

Сравнительный анализ структур информационных потоков и парка распределения персональных компьютеров в органах управления зачастую выявляет наличие дисбаланса между объемами информационных потоков и оснащенностью техникой отдельных служб.

1.3 Используемое программное обеспечение

В региональных администрациях используется различное общесистемное и специальное программное обеспечение для решения следующих задач:

- Операционные системы серверов и рабочих станций.
- Внешние Интернет–сервера (в первую очередь веб-сервера и почтовые сервера).
- Офисные программы.
- Программы для работы с графикой и издательской деятельности.
- Бухгалтерские и финансовые программы.
- Системы автоматизации делопроизводства и электронного документооборота.
- Геоинформационные системы.

Если в качестве серверной платформы (особенно Интернет-серверов) продукты с открытым кодом достаточно распространены, то на рабочих местах пользователей и в прикладных системах они в настоящее время встречаются крайне редко.

1.4 Подготовка кадров

Сотрудники государственной власти и местного самоуправления, в основном, имеют навыки работы со специализированным программным обеспечением на уровне пользователей (в соответствии со специальностью) и программным обеспечением Microsoft Office различных модификаций.

Доля специалистов по программному обеспечению с открытым кодом в органах власти весьма мала.

1.5 Текущее состояние ИТ-инфраструктур

Сейчас уже ни у кого не вызывает сомнения необходимость проведения работ в указанных направлениях использования ИКТ и обеспечение приоритетов в использовании информационных технологий. Так, во многих региональных администрациях разработаны и эксплуатируются информационные сайты, предоставляющие широким слоям населения информацию о разнообразных сторонах жизни города и области, деятельности органов власти. Внедряются в эксплуатацию системы электронного документооборота, проделана значительная работа по автоматизации управленческой и учетной деятельности органов власти, ведутся работы по сбору и обработке геоинформационных данных, закладываются основы для создания единой информационной системы.

В органах государственной власти широко используется Интернет и электронная почта. Однако в силу ряда причин имеющаяся ИТ-инфраструктура в настоящее время не удовлетворяет современным требованиям. Сохраняется существенная разница в уровнях информатизации и использования ИТ-услуг между центром области и низовыми звеньями системы управления областью (некоторые муниципальные образования, сельские органы власти). Базы данных имеют локальный характер, часто дублируют друг друга, а в ряде случаев и противоречат друг другу. Существенным недостатком являются значительные лицензионные отчисления и затраты на поддержание лицензионного программного обеспечения.

Существующее в настоящее время аппаратное обеспечение ИТ-инфраструктуры администраций представляет собой совокупность разнородных средств, часть из которых устарела как морально, так и физически, что в ряде случаев не позволяет установить современное программное обеспечение.

Сетевая инфраструктура областей использует, в основном, существующие линии телефонной связи. В поселках городского типа и сельских населенных пунктах современные аппаратные средства зачастую отсутствуют, а квалификация персонала поселковых и сельских администраций не соответствует современным требованиям.

Принципиальными являются меры, способные установить приемлемый уровень информационной безопасности и заложить возможности для администрирования сетевой инфраструктуры:

- Использование технологии сегментирования пользователей на рабочие группы по принадлежности к задачам управления.
- Обеспечение централизованной авторизации пользователей на портах доступа в сеть. Использование централизованного сервера для авторизации администраторов при доступе к консоли управления активным оборудованием. Обеспечение возможности запрета доступа в сеть для несанкционированных сетевых адаптеров.
- Фильтрация несанкционированного трафика на основе внутренней политики, принятой в администрации.

Существуют проблемы организационно-технического характера:

- неполнота исходных данных и ограниченный уровень информационной целостности в смежном ПО;
- недостаточный уровень автоматизации источников информации, прежде всего на уровне муниципальных, поселковых и сельских структур;
- недостаточный уровень вычислительных ресурсов, что снижает общую производительность и устойчивость функционирования системы;
- недостаточное развитие информационно-сетевой инфраструктуры.

В настоящее время используются программные разработки, носящие локальный характер, которые автоматизируют отдельные функции и задачи хозяйственной деятельности администрации. Выполненные по форме и технологии локальных автоматизированных рабочих мест, данные задачи реализованы собственными силами администрации и внешними подрядчиками.

В состав программного обеспечения входят разнородные средства, базирующиеся, в основном, на платформе Windows различных модификаций.

Информационное обеспечение представлено разрозненными локальными базами данных. Локальный характер выполняемых с их помощью задач, поддержания и заполнения приводит к отсутствию полноты, дублированию, противоречивости информации, что затрудняет их использование при принятии оперативных управленческих решений. Кроме того, локальная форма реализации задач приводит к увеличению трудоемкости сбора, консолидации информации (повторный ввод, бумажные носители) и уменьшает возможности для анализа и планирования. Отсутствие единой функционально и информационно интегрированной системы, покрывающей функции различных блоков управления, определяет большие периоды актуализации управленческой и учетной

информации. По ряду направлений управленческой деятельности базы данных не сформированы. Отсутствие централизованного высоконадежного хранения данных, средств централизованного резервного копирования повышает риск потери данных и краха системы при возникновении различного рода аварий и сбоев как на отдельных рабочих местах, так и в системе в целом.

Действующие в администрациях технологии сопровождения и поддержки ИТ-инфраструктуры создавалась эволюционно и зачастую не имеют ярко выраженной процессной направленности. Существующая технология сводится к устранению проблем по локальным зонам ответственности служб сопровождения с недостаточно налаженным между ними уровнем взаимодействия.

Для оперативного устранения выявленного инцидента существует несколько точек входа в виде списка контактных телефонов технических служб. При проявлении проблемы методологического или программно-технического характера у конечного пользователя он по своему усмотрению транслирует выявленный инцидент одной из перечисленных служб сопровождения в соответствии с индивидуальными представлениями о природе проблемы. При этом факт эскалации и устранения проблемы не фиксируется.

Наиболее перспективными направлениями совершенствования технологического обеспечения сопровождения и поддержки ИТ-инфраструктуры являются следующие:

1. Оперативная регистрация инцидентов с единой точкой входа, накопление соответствующей статистики с последующим анализом выявленных проблем;
2. Разработка и утверждение норм и правил обслуживания средств программного обеспечения, компьютерной и телекоммуникационной техники;
3. Ведение регистрации и статистики инцидентов по их количеству, характеру и времени устранения.

Исторический ход развития ИТ-инфраструктур администраций происходил эволюционно, путем добавления дополнительных аппаратных и программных средств по мере возникновения необходимости. Такое развитие ИТ-инфраструктуры зачастую приводит к ее лавинообразному усложнению, в результате чего резко возрастают расходы на ее поддержание и сопровождение при одновременном снижении эффективности управления. Дальнейшее развитие ИТ-инфраструктуры в таком направлении должно неизбежно привести к деградации системы, а приобретение лицензионного программного обеспечения и постоянное обновление лицензий — к значительным накладным расходам.

Как показывает зарубежный и отечественный опыт, единственным выходом из создавшейся ситуации является комплексная модернизация ИТ-инфраструктуры с заменой морально и физически устаревших аппаратных и программных средств на современные образцы, создание однородной современной аппаратной и программной базы, на основе которой и развернуть ИТ-инфраструктуру с широким применением программного обеспечения с открытым кодом и переносом на современную платформу специализированного программного и информационного обеспечения.

Глава 2

Модернизация ИТ-инфраструктуры

Очевидно, что в силу сложности задачи, отсутствия готовых решений процесс модернизации ИТ-инфраструктуры должен проводиться в несколько этапов с тщательным исследованием результатов, полученных на каждом этапе, и тщательным планированием последующих этапов. Кроме того, на некоторых этапах, особенно на начальных, возможно, потребуется проведение НИОКР, по результатам которых могут быть приняты стратегические решения.

Можно выделить некоторые из основных перспективных направлений первого этапа модернизации ИТ-инфраструктур региональных администраций. Такими направлениями являются:

1. широкое применение программного обеспечения с открытым кодом;
2. создание и широкое использование информационно-коммуникационных порталов как средства повышения информационной открытости органов власти и создания механизмов общественного контроля их деятельности;
3. создание и широкое использование интернет-порталов как средства сопряжения компонент ИТ-инфраструктуры;
4. создание системы электронного документооборота на основе интернет-порталов;
5. создание унифицированного комплекса программных средств обеспечения рабочего процесса должностных лиц органов управления;
6. внедрение геоинформационных систем;
7. применение GRID-технологий в процессах государственного управления.

В силу сказанного выше, на первом этапе проведения процесса модернизации целесообразно предварительно провести НИОКР — создать опытный образец ИТ-инфраструктуры, исследовать его совместимость с существующими технологиями управления, после чего при положительном результате — приступить к полномасштабному внедрению в существующую ИТ-инфраструктуру.

Ключевой задачей при проведении такой модернизации является создание информационного базиса. Информационный базис органов управления, осуществляющих анализ и подготовку управленческих решений, создается в виде единого информационного пространства региона. Структура информационного базиса включает документальную информацию, организованную в виде системы файлов на машинных носителях, а также систему из критериев и показателей (индикаторов), которые выбираются в соответствии с оперативными и долгосрочными целями органов управления. Информационный базис создают региональный центр информатизации и информационно-аналитический центр.

Основными источниками информации являются:

- управления статистики (экономико-статистическая информация);
- бюро технической инвентаризации (первичные данные об объектах недвижимости);
- управление архитектуры (первичная информация о планировках, данные градостроительства, характеристики участков земли);

- налоговые службы, службы регистрации юридических лиц, банковские системы (паспорта юридических лиц);
- центральные адресные бюро, ЗАГСы, службы социального обеспечения, медицинские страховые компании (персональные данные о населении).

Кроме того, непосредственными источниками являются:

- федеральные органы законодательной и исполнительной власти;
- административные органы местного самоуправления;
- администрации соседних регионов;
- региональные и межрегиональные объединения, ассоциации;
- государственные, коммерческие и общественные организации.

Основой функциональной архитектуры информационно-прикладного уровня интегрированной системы должно стать единое хранилище данных и документов, являющееся информационным базисом органов управления и реализованное с применением современных технологий накопления и обработки больших объемов данных, хранящихся в электронном виде.

Единое информационное хранилище должно содержать формализованную и неформализованную информацию как организационно-экономического содержания, так и учетно-отчетного, накапливаемую в процессе эксплуатации функциональных информационных систем, входящих в состав ИАСУ.

Основными принципами построения информационной архитектуры ИАСУ являются:

- унификация средств централизованного хранения и обработки данных;
- реализация открытых интерфейсов обмена информацией между различными элементами информационной архитектуры;
- обеспечение информационной совместимости с внешними информационными системами Федеральных и местных органов власти, банков, поставщиков и других партнеров;
- консолидация информации в базах данных, реализуемых по функциональному принципу с учетом расположения источников информационного наполнения;
- информационная совместимость с применением следующих правил:
- надежность и информационная безопасность — ИАСУ должна содержать процедуры обеспечения надежности и безопасности данных для всех элементов информационной архитектуры;
- создание универсальной сетевой среды доступа пользователя к централизованной и, при необходимости, распределенной информации с возможностью просмотра графического образа.

Формирование единого информационного пространства предполагает селекцию включаемой в него информации по достоверности, полезности, новизне и требует соблюдения ряда взаимоувязанных принципов: полноты, временной детерминированности, доступности, открытости, иерархичности, прогностичности, обратной связи.

Принцип полноты связан с охватом всех информационных источников по всем основным направлениям деятельности органов управления (регулярные статистические данные о ходе реформ, результаты выборочных обследований и опросов по всему спектру проблем, связанных с развитием региона, мнения экспертов, контент-анализ прессы, документы общественных организаций и т.д.). Источники должны обеспечивать достаточно полную, адекватную оценку ситуации и, в первую очередь, включать такие, которые позволяют отслеживать быстро протекающие процессы на отдельных наиболее важных и/или типичных для региона объектах.

Принцип иерархичности позволяет рассматривать ход реализации принятых решений на всех уровнях управления, в различных сферах народного хозяйства и их влияние на социально-экономическую ситуацию.

Временная детерминированность определяет для каждого информационного сообщения свою периодичность формирования и представления (неделя, месяц, год, и т. д.). Поэтому структура информационного базиса должна учитывать частоту обращений и характер поступающих на документы запросов.

Принцип прогностичности предлагает фиксирование не столько отдельных фактов, событий, но, в первую очередь, выявление характера тенденций в развитии региона и формирование предложений по их усилению или ослаблению.

Соблюдение принципа открытости обеспечивает быстрое включение в информационный базис показателей и критериев, специфичных для конкретных районов или складывающейся социально-экономической ситуации. В связи с этим появление и исчезновение показателей не должно оказывать заметного влияния на структуру информационного базиса.

Принцип доступности определяет возможность получения данных, необходимых для оценки того или иного фактора, для расчета того или иного показателя в течение времени, ограниченного принятием решения на всех уровнях управления.

Информационное пространство должно обеспечивать действенную обратную связь между руководством, принимающим решения, и структурными подразделениями, где должны реализовываться принятые решения. С этой целью должна проводиться специальная оценка результативности представленной информации по эффективности принятого решения и соответствующие изменения информационного базиса.

С целью предоставления актуальной информации пользователям ИАСУ в режиме on-line с учетом их географического распределения техническая реализация корпоративного хранилища данных и документов ИАСУ должна выполняться на базе центра консолидации вычислительных ресурсов: полнофункционального Центра хранения и обработки данных.

Информационная взаимосвязь между потребителями информации и актуализация централизованной информации осуществляется в сеансовом режиме пакетной репликации данных по модернизированным магистральным каналам между потребителями.

Схема использования вычислительных ресурсов (централизованная или распределенная) и условия актуализации данных определяются индивидуальными особенностями применения каждой функциональной ИС в отдельности.

2.1 Базовое программное обеспечение (БПО)

Этот слой программного обеспечения назван базовым потому, что оно является основой, базой для построения всей программной части ИТ-инфраструктуры. Базовое ПО включает в себя набор программных средств, которые используются постоянно. Их конкретный состав может различаться в зависимости от сферы применения. Основными компонентами, входящим в состав большей части таких наборов, являются ядро операционной системы, базовый набор системных утилит. Для снижения совокупной стоимости владения рекомендуется проведение унификации используемого набора ПО путем создания единого набора БПО, расширяемого в зависимости от конкретных сфер применения. Имеющаяся ИТ-инфраструктура включает серверные и пользовательские компоненты, что потребует создания унифицированных решений для каждой из указанных областей. Серверное БПО дополняется набором программных средств, поддерживающих стандартные Интернет/Интранет сервисы, системами резервирования ресурсов, средствами администрирования программно-аппаратного комплекса. БПО пользовательского уровня должно включать в себя графическое окружение пользователя, набор средств для работы с Интернет/Интранет ресурсами и офисный пакет, обеспечивающий возможность обмена документами с другими системами.

При выборе базового программного обеспечения следует руководствоваться следующими критериями оценки:

- минимизация лицензионных издержек;
- возможность сквозного аудита исходных кодов и проведения процесса их пересборки для получения гарантии отсутствия «программных закладок»;
- соответствие распространенным стандартам открытых систем (POSIX, BSD, SYSV, LSB);
- устойчивость к вирусным атакам и вторжениям хакеров;
- наличие достаточного набора дополнительного программного обеспечения;
- возможность использования исходных кодов системы для дополнения их новой функциональностью и создания на их базе новых программных продуктов;
- наличие местного штата специалистов, владеющих данными технологиями.

Данным требованиям в наибольшей степени соответствуют системы, построенные на базе ОС GNU/Linux. Успешность внедрения таких технологий во многом зависит от наличия местных групп пользователей Linux

(LUG — Linux User Group) и центров компетенции Linux, коммерческих компаний, занимающихся разработкой, внедрением и поддержкой таких решений.

2.2 Информационно-коммуникационные средства

В соответствии с требованиями Концепции использования информационных технологий в деятельности федеральных органов государственной власти до 2010 года одним из приоритетов является «Повышение информационной открытости деятельности федеральных органов государственной власти, доступности соответствующей информации для граждан и организаций, а также создание механизмов общественного контроля их деятельности».

Информационным и функциональным ядром средств реализации принципа открытости органов власти, информирования населения и обеспечения связи населения и органов и представителей власти должны стать информационно–коммуникационные порталы. Отличительной особенностью перспективного решения информационно-коммуникационных средств является включение в состав внедряемой функциональности дополнительных типовых возможностей обеспечения участия населения в управленческой деятельности путем организации обратной связи между органами и представителями власти и широкими слоями населения, четкого и централизованно управляемого разграничения прав доступа к ресурсам, увеличение возможностей по контролю за достоверностью публикуемой информации (использование workflow), а также возможность интеграции их функциональности с Системой управления документооборотом.

Информационно-коммуникационные средства должны строиться по централизованной схеме обработки данных и их коллективного использования.

Таким образом, в рамках реализации требования открытости органов власти в ходе модернизации ИТ-инфраструктуры должна быть реализована система информационно-коммуникационных порталов, предназначенных для:

- освещения деятельности администрации области и правительства Российской Федерации путем размещения в нем нормативных актов, документов и иной информации и предоставления возможности населению области ознакомиться с этими документами в удобной для восприятия форме;
- осуществления обратной связи администрации области с населением путем предоставления возможности населению размещать в разделах портала письма, запросы, обращения и т.п. к должностным лицам администрации и опубликования ответов должностных лиц на обращения и запросы граждан.

В порталах должны размещаться законодательные акты, решения, постановления, документы и иная информация, предназначенные для широкого информирования граждан области. Для структурированного размещения таких документов порталы должны содержать тематические разделы и средства для размещения документов в таких разделах. Для доступа и поиска информационных материалов порталы должны предоставлять удобные, интуитивно понятные средства (поисковые системы, средства навигации и т.д.).

Порталы должны предоставлять средства для обеспечения возможности гражданам выражать свое мнение по различным вопросам, обращения к должностным лицам различных уровней с письмами и заявлениями и размещения ответов должностных лиц на заданные жителями вопросы.

Основными целями создания информационно-коммуникационных порталов администрации Новгородской области и муниципальных образований являются:

- информирование граждан, общественных организаций и других пользователей о деятельности органов управления администрации области и других государственных учреждений путем предоставления свободного доступа с использованием современных компьютерных технологий к электронным информационным базам администрации области;
- обеспечение постоянного поддержания электронных информационных баз администрации области в актуальном состоянии;
- повышение качества принимаемых решений, сокращение сроков их подготовки за счет оперативного учета мнения граждан области;
- повышение эффективности законотворческой, управленческой деятельности и исполнительности в исполнительных и законодательных органах муниципального образования, в организациях и на предприятиях

области за счет улучшения информированности, облегчения и упрощения доступа руководства и исполнителей к информационным базам администрации области, а так же за счет полноты и достоверности хранимых документов;

- снижение социальной напряженности среди населения, вызванной неинформированностью населения о деятельности администрации области.

2.3 Пользовательское ПО общего назначения

К данной категории систем относятся программные решения, обеспечивающие создание информационного окружения для участников функциональных ИС и их межсистемные взаимодействия в рамках задач ИАСУ области. В этом разделе будут рассмотрены:

- инфраструктура рабочих мест;
- система управления документооборотом.

Система управления документооборотом по принятой международной классификации относятся к классу систем управления документами, построенных на концептуальной основе, в центре которой находится документ. При этом особенность систем управления документооборотом заключается в том, что они оперируют структурированными данными.

Комплекс программных средств «инфраструктура рабочих мест» предназначен для осуществления информационной и функциональной поддержки деятельности участников процесса управления областью. К данной категории можно отнести широко используемые типовые программные пакеты офисного назначения, средства обеспечения работы функциональных и унаследованных приложений.

2.4 Инфраструктура рабочих мест

Инфраструктура рабочих мест — это унифицированный набор технологий, программных и аппаратных средств, обеспечивающих работу должностных лиц с различными автоматизированными системами и информационными ресурсами региона.

На сегодняшний день инфраструктура рабочих мест состоит из большого количества (несколько сотен) персональных компьютеров на платформе Intel с различными техническими параметрами, многообразного периферийного оборудования (принтеры, сканеры и т. п.), а также совокупности программных средств (различные операционные системы семейства Microsoft Windows, офисные приложения семейства Microsoft Office разных версий и другие приложения).

Описанная инфраструктура порождает ряд проблем:

1. Для работы с программным обеспечением требуется приобретение лицензий;
2. Каждое новое поколение базовых программных средств (операционная система, стандартные офисные приложения) требует приобретения новых персональных рабочих станций без возможности целесообразного использования заменяемого оборудования;
3. Защита информации (как в части возможностей нелегального доступа, так и в части сохранности информации) организована неоптимально;
4. Обслуживание всей инфраструктуры рабочих мест является сложной задачей, что приводит к неоптимальной ее эксплуатации и росту затрат на управление.

При выборе программного обеспечения для проведения модернизации следует руководствоваться следующими критериями:

1. Минимизацией лицензионных отчислений на программное обеспечение;
2. Высокую степень информационной безопасности и надежности;
3. Наличие эргономичного русскоязычного пользовательского интерфейса;
4. Наличие набора приложений, позволяющих полноценно работать с Интернет/Интранет серверами (WEB, E-Mail);

5. Наличие офисных приложений для работы с офисными документами;
6. Наличие средств интеграции с унаследованными приложениями;
7. Возможность создания специализированных пользовательских окружений для решения профессиональных задач;
8. Возможностью удаленного администрирования (в том числе и по медленным каналам связи) удаленных рабочих мест;
9. Поддержка распространенных сетевых протоколов (X-Window, RDP, VNC) для обеспечения возможности удаленного запуска графических приложений и поддержки технологии «тонкий клиент».

Одним из способов оптимизации затрат, повышения качества и ослабления зависимости от конкретных поставщиков является построение Инфраструктуры рабочих мест на основе программного обеспечения с открытым кодом (Open Source).

Для тиражирования и отчуждения от конкретного поставщика услуг решений и технологий, примененных в ходе построения Инфраструктуры рабочих мест, предлагается создать специализированный типовой комплект программных компонентов, эксплуатационной и методической документации. Такой комплект может использоваться при дальнейшей модернизации Инфраструктуры рабочих мест в органах управления, в том числе и в иных регионах.

Образец инфраструктуры рабочих мест должен обеспечивать выполнение следующих основных функций:

1. получение и отправление электронных писем, ведение архива собственной переписки, ведение расписаний;
2. работа с документами, электронными таблицами и презентациями (в том числе в форматах Microsoft Office);
3. использование имеющихся информационных систем в рамках выполнения своих должностных обязанностей;
4. реализация собственного информационного окружения по идеологии «рабочего стола»;
5. работа с принтерами, сканерами, факсами и другими периферийными устройствами персонального либо коллективного пользования.

Для информационных систем с веб-доступом в качестве средства доступа должен применяться интернет-браузер. Для информационных систем, имеющих клиентские части, выполняемые на рабочем месте, должен использоваться эмулятор ОС Windows, либо работа с такими приложениями должна быть организована в терминальном режиме

Инфраструктура рабочих мест должна использовать различные варианты модели вычислений и обеспечивать возможность доступа должностных лиц к автоматизированным системам, работающим на платформах семейства Microsoft Windows:

- Полноценное рабочее место, являющееся средой выполнения различных клиентских приложений. Технические требования, предъявляемые к такому рабочему месту, зависят от количества и назначения выполняемых на нем приложений и являются, обычно, достаточно высокими.
- Различные технологии использования рабочего места в качестве «тонкого» клиента. Эти технологии позволяют существенно снизить технические требования к рабочим станциям за счет ограничения набора выполняемых на них приложений. Данные технологии позволяют также не увеличивать требования к пропускной способности каналов связи путем оптимизации объемов передаваемых данных.

Использование программного обеспечения с открытым кодом (в первую очередь Linux-систем) в качестве основы Инфраструктуры рабочих мест, обладает следующими преимуществами:

1. все предлагаемые программные компоненты обладают нулевой стоимостью лицензий на использование;
2. в составе предлагаемых решений имеются развитые средства управления (в том числе удаленного) программными компонентами, развернутыми на рабочих станциях, что позволяет администратору системы гибко и оперативно конфигурировать рабочие места в соответствии с функциональными потребностями оператора рабочего места;
3. возможные варианты модели вычислений позволяют оптимально распределять вычислительную нагрузку между рабочими станциями и серверами, входящими Центр обработки данных;

4. применением модели вычислений «тонкий клиент» обеспечивается возможность дальнейшего использования морально устаревших компьютеров, которые уже не могут эксплуатироваться в качестве полноценных рабочих станций;

5. используемая операционная система Linux обеспечивает высокую степень информационной безопасности и надежности;

6. обслуживание рабочих станций, работающих под управлением операционной системы Linux, может выполняться удаленно, в том числе, с использованием низкоскоростных каналов связи.

Для тиражирования и отчуждения от конкретного поставщика услуг решений и технологий, примененных в ходе построения Инфраструктуры рабочих мест, рекомендуется создавать специализированный типовой комплект программных компонентов, эксплуатационной и методической документации. Такой комплект может использоваться при дальнейшей модернизации Инфраструктуры рабочих мест в органах управления, в том числе и в иных регионах.

2.5 Системы электронного документооборота

Тема повышения эффективности деятельности органов государственной власти и управления на основе широкого использования информационно-коммуникационных технологий (ИКТ), а также перевода взаимоотношений государства с обществом на уровень, соответствующий требованиям информационного общества, в последнее время неоднократно освещалась в информационных материалах государственных органов и СМИ.

В этом контексте, бесспорно, одним из приоритетных направлений модернизации ИТ-инфраструктуры органов государственной власти является развитие систем электронного документооборота, обеспечивающих технологическую основу для эффективной деятельности органов государственного управления, созданию открытого гражданского общества, сокращению неэффективных затрат на госсектор, улучшению взаимодействия органов государственной власти с гражданами и хозяйствующими субъектами.

Интенсивное развитие законодательной базы в области использования ИКТ, происходящее в последнее время, в свою очередь, создает благоприятные условия и правовую основу для внедрения электронного документооборота в органы государственной власти и управления.

Очевидно, что помимо классических задач, решаемых современной СЭД — организация создания и хранения электронных документов, а также управление их жизненным циклом, при разработке СЭД необходимо заложить основу для решения задач, связанных с эффективным информационным обеспечением деятельности сотрудников организации и управлением деловыми процессами.

Отличительными особенностями электронного документооборота государственных учреждений являются повышенные требования к информационной безопасности, а также обязательное установление авторства электронных документов и обеспечение их юридической значимости.

Положительный мировой опыт свидетельствует, что работу по внедрению СЭД в государственных органах рекомендуется начинать с небольшого пилотного проекта с целью выявить эффект от внедрения данной системы и принять решение о ее выборе, выработать базис для составления проекта реальной системы, составить полный план работ по разработке и внедрению.

2.6 Информационные порталы

Информационный портал должен обеспечивать сопряжение инфраструктуры рабочих мест с иными компонентами региональной ИТ-инфраструктуры. Портал выступает в качестве базового и унифицированного пользовательского и/или программного интерфейса, позволяет создать единую точку входа для работы с приложениями с общей политикой авторизации и разграничения прав доступа, единым пользовательским интерфейсом, возможностью регулирования набора предоставляемых сервисов. Это дает возможность использования прикладного программного обеспечения из единого репозитория как для сотрудников органов управления, так и для населения.

В рамках портальных технологий имеются средства быстрой разработки приложений (Rapid Application Development), с помощью которых может эффективно выполняться интеграция приложений и организовываться информационное обслуживание пользователей с поддержкой их должностной специфики. Кроме того,

технологическая платформа порталов поддерживает набор функций управления потоками работ (Work Flow) и электронного документооборота.

Основные критерии оценки портальных технологий для использования в ИТ-инфраструктуре:

1. возможность доступа через Internet/Intranet;
2. соответствие пользовательского интерфейса основным требованиям по доступности для различных групп пользователей (в том числе с ограниченными физическими способностями) и технологическим стандартам на веб-интерфейсы (возможность просмотра любым пользователем);
3. возможность интеллектуального поиска и индексирования информации, в том числе по дополнительной метаинформации, связанной с объектами;
4. возможность интернационализации содержимого сайта и его интерфейсов при предоставлении внешнего доступа к отдельным областям портала;
5. встроенные механизмы средств авторизации и разграничения прав доступа, единые для всех компонент портала;
6. возможность доступа к объектам портала в абстракции файловой структуры через стандартные сетевые протоколы (FTP, WEBDAV и т.п.);
7. наличие стандартных методов обмена информацией о текущих изменениях информационного наполнения и состояния объектов портала с приложениями инфраструктуры рабочих мест (SMTP, POP, IMAP, RSS, iCal и т.п.);
8. четкое разделение программного уровня, уровней информационного наполнения и представления информации;
9. поддержка концепции рабочего цикла (Work Flow), контролирующего логику изменения состояния объектов, помещенных в портал;
10. наличие упрощенного интерфейса управления основными параметрами портала и содержащихся в нем объектов с доступом через WEB;
11. возможность программного сопряжения портала с другими модулями и комплексами в соответствии со стандартными спецификациями (XML-RPC, SOAP и т.п.);
12. возможность доступа к внешним источникам данных (базы данных, сервера каталогов, активные источники данных и т.д.);
13. возможность построения портального решения на принципах программного обеспечения с открытым кодом (Open Source).

В современной ИТ-индустрии доминирующее влияние занимают следующие технологические платформы для создания распределенных вычислительных сред и разработки интеграционных модулей:

- LAMP (Linux, Apache, MySQL/PostgreSQL, PHP/Perl/Python)
- J2EE
- .NET

Для решения задач модернизации региональной ИТ-инфраструктуры могут быть использованы различные технологические платформы информационных порталов, интеграции информационных ресурсов, быстрой разработки приложений на основе специализированных программных продуктов с открытым кодом (Open Source):

1. PHP
2. Python, Zope и Plone.
3. Java

2.7 Геоинформационные системы

Применение ГИС-технологий для органов государственного управления ставит, в первую очередь, проблему выбора соответствующего программного и аппаратного обеспечения. Трудность заключается в том, что в на-

стоящее время не существует законченных отечественных разработок, полностью удовлетворяющих ставящейся задаче. Программное обеспечение геоинформационной системы должно обладать набором функциональных возможностей, характерных для данного класса программных средств:

1. ввод данных в машинную среду;
2. конвертирование цифровых данных для экспорта и импорта;
3. программные средства преобразования систем координат и трансформации картографических проекций;
4. хранение и управление позиционными и непозиционными атрибутами в базах данных;
5. растрово-векторные операции;
6. измерительные операции;
7. полигональные операции;
8. аналитические и моделирующие операции;
9. анализ поверхностей;
10. вывод данных и документированных результатов;
11. цифровая обработка дистанционных изображений.

Существует три варианта выбора программного обеспечения:

1. разработка нового программного продукта;
2. выбор и доработка отечественного программного продукта;
3. приобретение зарубежного комплексного программного продукта.

Первый вариант требует больших затрат времени и коллектива специалистов, имеющего опыт создания подобных систем. Ни тем, ни другим область на данный момент не располагает.

Большинство существующих отечественных разработок в области ГИС-технологий начинались как узкие приложения и их доработка непредсказуема как по стоимости, так и по результатам. Практические результаты в основном были получены только в оборонном комплексе, однако специфика разработки и применения, специализированные вычислительные комплексы и соответствующее программное обеспечение к ним не позволяют использовать их при создании ГИС в целях регионального управления.

Применительно к третьему варианту изучение опыта других регионов и организаций России показывает, что в работах по созданию ГИС в большинстве случаев использовались именно зарубежные программные средства. Использование данных программных средств (ПС) имеет ряд привлекательных черт, тем не менее, использование зарубежных программных продуктов ни в коей мере не должно привести к отказу от использования и развития отечественных разработок, особенно по тем направлениям, где полученные результаты превосходят зарубежные аналоги.

К недостаткам использования зарубежных ПС следует отнести необходимость проведения более жестких мероприятий по созданию комплексной системы защиты данных, циркулирующих в ГИС.

Геоинформационная система органов государственного управления (ГИС ОГУ) представляет собой многоуровневую иерархическую структуру. Основной структурной единицей является объект ГИС ОГУ, представляющий собой программно-технический комплекс государственного управления. Каждый объект ГИС ОГУ обязан обеспечивать необходимой информацией органы управления своего уровня, а также поставлять для вышестоящих объектов обобщенную информацию в объеме, отвечающем уровню ответственности.

Объекты ГИС ОГУ представляют собой программно-технические комплексы, развертываемые при организациях, входящих в состав органов государственного управления области, а также подчиненных им органов.

На уровне субъекта Федерации объекты ГИС ОГУ можно разделить на две группы, в зависимости от возложенных функций и положения в структуре управления:

1. объекты ГИС ОГУ сбора, накопления, обработки и выдачи первичной информации в виде тематических слоев. Здесь же осуществляется привязка всей совокупности объектов инфраструктуры к территории. К данным объектам ГИС ОГУ относятся ГИС объектного и территориального уровней;

2. объекты ГИС ОГУ принятия управленческих решений на основе обобщения данных, их анализа и представления в удобной пользователю форме. К данным объектам ГИС ОГУ относятся ГИС регионального уровня.

Такое разделение (выход — ГИС организации, вход — ГИС администрации) позволяет разгрузить уровень принятия управленческих решений от ответственности за полноту, достоверность и актуальность первичной информации и сосредоточиться на поиске оптимальных решений по конкретной проблеме.

Объект ГИС ОГУ представляет собой унифицированный набор функциональных компонент:

1. пользовательский интерфейс;
2. СУБД;
3. ввод данных;
4. базы данных;
5. средства анализа;
6. отображение и генерация конечного продукта.

Содержание и форма выходной информации (таблицы, диаграммы, графики, картографические документы и т.п.) определяются потенциальным пользователем на этапе проектирования ГИС.

Объекты ГИС ОГУ взаимодействуют путем обмена информацией. Объекты верхнего уровня ГИС ОГУ получают необходимую информацию от объектов нижнего уровня. Объекты нижнего уровня должны быть обеспечены нормативно-справочной информацией, поступающей от верхних объектов.

Периодичность обмена данными и способ их передачи определяются их типом. Данные, передаваемые от объектов нижнего уровня на объекты верхних уровней, могут быть разделены на три основных типа:

1. долговременного использования;
2. оперативные сведения;
3. срочные сообщения.

К информации долговременного использования могут относиться сведения о запасах, состоянии природных ресурсов, населении региона и т.п. Данные этого типа могут актуализироваться и передаваться объектам высшего уровня через установленные (достаточно большие) промежутки времени.

К оперативным сведениям относятся сводки о состоянии быстро протекающих процессов, например, о ходе посевной кампании, реализации промышленных товаров и т.п. Такие сведения должны передаваться по мере их накопления и обобщения.

К срочным относятся сообщения, требующие немедленного принятия решений в органах власти, например, данные о крупной аварии, стихийном бедствии, массовых беспорядках и т.п.

Взаимоотношения между объектами ГИС ОГУ основываются на законодательстве Российской Федерации и нормативных и распорядительных актах, принятых Главой администрации и Думой области.

Организационно-координирующая роль в создании и функционировании ГИС ОГУ области (в т.ч. и вопросы взаимодействия объектов ГИС ОГУ) должна принадлежать региональному центру информатизации (РЦИ). Осуществляя функцию регистрации ГИС и их анализа, он может давать рекомендации по проведению целенаправленной политики развития ГИС ОГУ области и быть инициатором разработки и принятия соответствующих нормативных документов, в т.ч. определяющих порядок, сроки, конкретный состав и частоту передачи данных между объектами ГИС ОГУ области.

Для обоснованного принятия решений на высшем уровне органов государственного управления области необходима исчерпывающая информация, в том числе об объектах всех уровней ведения. Поэтому информация, отнесенная к ведению области или собранная за счет бюджетных средств, должна поступать со всех объектов ГИС ОГУ постоянно, оперативно и безвозмездно.

Функционирование и взаимодействие отраслевых и территориальных объектов ГИС ОГУ (в т.ч. организаций-генераторов, владельцев, посредников и пользователей информации) должны регулироваться на основе соглашений, договоров, контрактов и других документов, определяющих условия их взаимодействия, и строиться, в основном, на взаимовыгодной основе.

2.8 Применение GRID-технологий

Важными компонентами современных интегрированных решений следует признать промежуточные программные продукты (middleware), которые позволяют интегрировать вычислительные ресурсы в заданном масштабе.

Grid-технологии — это новый способ абстракции вычислительных ресурсов и ресурсов хранения, позволяющий обеспечить единообразный доступ к ресурсам организации.

Данная технология позволяет осуществлять крупные объемы вычислений с большими массивами данных с высокой степенью защищенности при использовании в качестве транспортного уровня сети публичного доступа Интернет. Этот подход в данный момент широко используется как в научных исследованиях, так и в коммерческих проектах.

Набольшее распространение получил программный пакет Globus Toolkit (http://www.globus.org). Этот проект развивается уже достаточно долгое время (с 1997 года) и хорошо зарекомендовал себя во многих крупных исследовательских проектах, в том числе и международных.

Наболее известные из них:

Large Hadron Collider Computing GRID (LCG) — http://lcg.web.cern.ch/lcg

Enabling Grids for E-sciencE (EGEE) — http://public.eu-egee.org

Earth System Grid (ESG) — https://www.earthsystemgrid.org/home/publicHomePage.do

Подробнее смотри: http://www.globus.org/alliance/projects.php.

Использование такой архитектуры при построении распределённой компьютерной системы даёт все необходимые предпосылки для интегральной продолжительной экономии финансовых ресурсов на развёртывание системы и создания защищенных доверенных вычислительных сред («виртуальных организаций») между различными субъектами организационно-хозяйственной деятельности, причем одна физическая организация может входить в несколько виртуальных. Экономия достигается, главным образом, благодаря возможности более эффективного использования дорогостоящего оборудования и квалифицированных кадров для решения различных задач и приложений независимо от их географического расположения.

Возможные сферы применения данной инфраструктуры:

1. Построение надежной системы обмена информацией между различными организациями, независимой от конкретных реализаций программных модулей, осуществляющих такой обмен.
2. Консолидация большого количества вычислительных ресурсов, входящих в виртуальную организацию, для проведения сложных расчетов с большим количеством вычислений и данных: аналитика, оптимизация, моделирование процессов, связанных с задачами управления.
3. Предоставление ресурсов по запросу от других государственных учреждений, учебных и научных организаций, промышленных предприятий.

Информационную систему организации, с точки зрения GRID-архитектуры, можно представить как некий совокупный вычислительный элемент и элемент хранения. Вычислительный элемент может служить единой входной точкой доступа к распределенным вычислительным ресурсам; тогда как элемент хранения обеспечивает распределенное хранилище данных с предоставлением унифицированного платформонезависимого интерфейса доступа к распределенному массиву данных. Доступ к каждому из этих элементов базируется на стандартизированном интерфейсе веб-сервисов, совместимую со стандартами консорциума W3 на WSDL и технологию SOAP, которые являются общепринятым стандартом коммуникации в гетерогенных вычислительных средах.

Единая схема доступа через интеграционный портал позволяет обратиться к любому сервису и документу в пределах распределенной информационной системы, при этом их конкретное местоположение несущественно, поскольку обеспечивается наличие сквозной системы имен.

Для контроля доступа в системе Globus используются механизм PKI-сертификатов и механизм виртуальных организаций. Это дает возможность обеспечить безопасную интеграцию данных, логически относящихся к различным организационным структурам.

Данные технологии можно рассматривать как набор перспективных механизмов взаимодействия и осуществления возможности интеграции вычислительных ресурсов и сервисов различных Информационно-ресурсных центров: региональных администраций, государственных органов, промышленных предприятий, учебных и научных заведений и др.

Глава 3

Модернизация системной инфраструктуры

3.1 Центр Обработки Данных (ЦОД)

В настоящее время и в обозримой ближайшей перспективе наиболее актуальной моделью построения системной части ИТ-инфраструктуры будет сетецентрическая модель, где все данные, сервисы и приложения хранятся и исполняются централизованным образом в специальной структуре Центра обработки данных (ЦОД), который обеспечивает единый, целостный информационный ресурс с гарантированными уровнями достоверности, доступности и безопасности данных.

Центр обработки данных является комплексным организационно-техническим решением, предназначенным для создания высокопроизводительной, отказоустойчивой информационной инфраструктуры и обеспечения единого, целостного информационного ресурса для структурных подразделений с гарантированными уровнями достоверности, доступности и безопасности данных.

ЦОД видится как вычислительная инфраструктура (набор взаимосвязанных программных и аппаратных компонентов, организационных процедур, мест локации и персонала), предназначенная для безопасной централизованной обработки, хранения и предоставления данных сервисов приложений.

Основные задачи ЦОД:

- эффективное консолидированное хранение и обработка данных;
- предоставление пользователям прикладных сервисов;
- обеспечение функционирования прикладных систем.

Создание ЦОД хорошо согласуется с общей стратегией модернизации ИТ-инфраструктуры, обеспечивает консолидацию и эффективное использование вычислительных ресурсов и данных, входящих в состав информационного обеспечения прикладных функциональных систем. ЦОД обеспечивает решение конкретных текущих и перспективных задач, порождаемых изменяющимися и возникающими административными процессами.

Достоверность и безопасность данных технически обеспечивается использованием мощных централизованных средств хранения и архивации, единой точкой контроля за доступом к приложениям и данным, защитой данных в ЦОД от несанкционированного доступа.

Организационные меры обеспечения информационной безопасности значительно эффективнее в случае консолидированного хранения данных. При этом разрабатываются единые стандарты эксплуатации и обслуживания ЦОД, физического доступа в ЦОД персонала, коммуникационного доступа к данным ЦОД через каналы связи.

Характерным свойством современных ЦОД является высокая степень управляемости ресурсами и, как следствие, максимальная полнота и контроль использования этих ресурсов.

ЦОД обеспечивает, с одной стороны, консолидацию технических и информационных ресурсов, с другой стороны, организацию эксплуатационного процесса для поддержания соответствующего уровня качества услуг, которые предоставляются в виде информационных сервисов.

В ЦОД реализуется комплексный подход к использованию ИТ ресурсов, к задачам обработки информации, что позволяет решить вопросы повышения надежности и стабильности, гибкости и масштабируемости ИТ

сервисов, их управляемости, возможности легкой модернизации используемого аппаратного и программного обеспечения.

Комплексные решения, применяемые в ЦОД, позволяют одновременно рассматривать такие важные аспекты предоставления ИТ услуг, как ценовая политика, решение сервисных вопросов, работа с критичными приложениями, упрощение администрирования, обеспечение защиты, режим работы при восстановлении после катастроф.

Предлагается решение для перечисленных задач за счет проектирования двух уровней ЦОД: физического размещения и логических взаимосвязей.

- Уровень физического размещения — это решения по физическому расположению компонентов ЦОД, структуре и составу инфраструктуры ЦОД (необходимых помещениях, объеме коммуникаций и т.д.).
- Уровень логических взаимосвязей определяет набор коммуникационных протоколов, которые будут использоваться в ЦОД, применение общесистемных платформ для управляющих сервисов.

При создании ЦОД следует обратить особое внимание на организацию следующих возможностей:

1. системной консолидации и интеграции различных ИТ ресурсов (серверов, данных, и т.д.);
2. управление всеми ресурсами из одной точки;
3. централизацию размещения, дистрибуции, доступа и управления для всех основных прикладных программных приложений;
4. обеспечение актуализации и достоверности данных и необходимого уровня информационной безопасности;
5. создание единой системы управляющих сервисов;
6. построение объединенных хранилищ данных;
7. использование различных платформ.

3.2 Модернизация БПТИ

Основные стратегические положения модернизации ИТ-инфраструктуры включают в себя ориентацию на интегрированные прикладные решения и преимущественное использование ПО с открытым кодом. Оба эти положения непосредственно определяют подходы к модернизации базовой программно-технологической инфраструктуры (БПТИ).

Для реализации первого положения в структуре БПТИ должна быть создана платформа, обеспечивающая интеграцию функций прикладных систем и информационных ресурсов. В качестве такой платформы предлагается использовать современные портальные технологии и средства, описанные в разделе Раз. Е.5.

Согласно второму положению при модернизации БПТИ определяется преимущественное использование ПО с открытым кодом. Имеющийся на сегодняшний день арсенал такого ПО охватывает все элементы БПТИ, начиная от прикладного ПО и заканчивая ОС и СУБД, включая специализированное ПО для мобильных устройств. Это позволяет создавать любые прикладные решения, необходимые для автоматизации деятельности органов государственной власти и местного самоуправления.

Вместе с тем, принимая во внимание большое количество ранее разработанных и используемых администрациями локальных систем и АРМ, необходимо учитывать следующие тактические соображения.

1. Вновь создаваемые прикладные решения могут и должны вестись в соответствии с декларированной стратегией использования базового ПО с открытым кодом.
2. Элементы прикладного программного обеспечения общего назначения (офисные пакеты, электронная почта и т.д.), входящие в состав рабочего окружения пользователя, должны замещаться постепенно, по мере обновления парка вычислительной техники, используемой на рабочих местах.
3. Для обеспечения постепенного перехода на новые программные средства БПТИ и поддержки существующих специализированных программных комплексов и АРМ БПТИ должна предусматривать возможность запуска и работы с этими унаследованными приложениями.

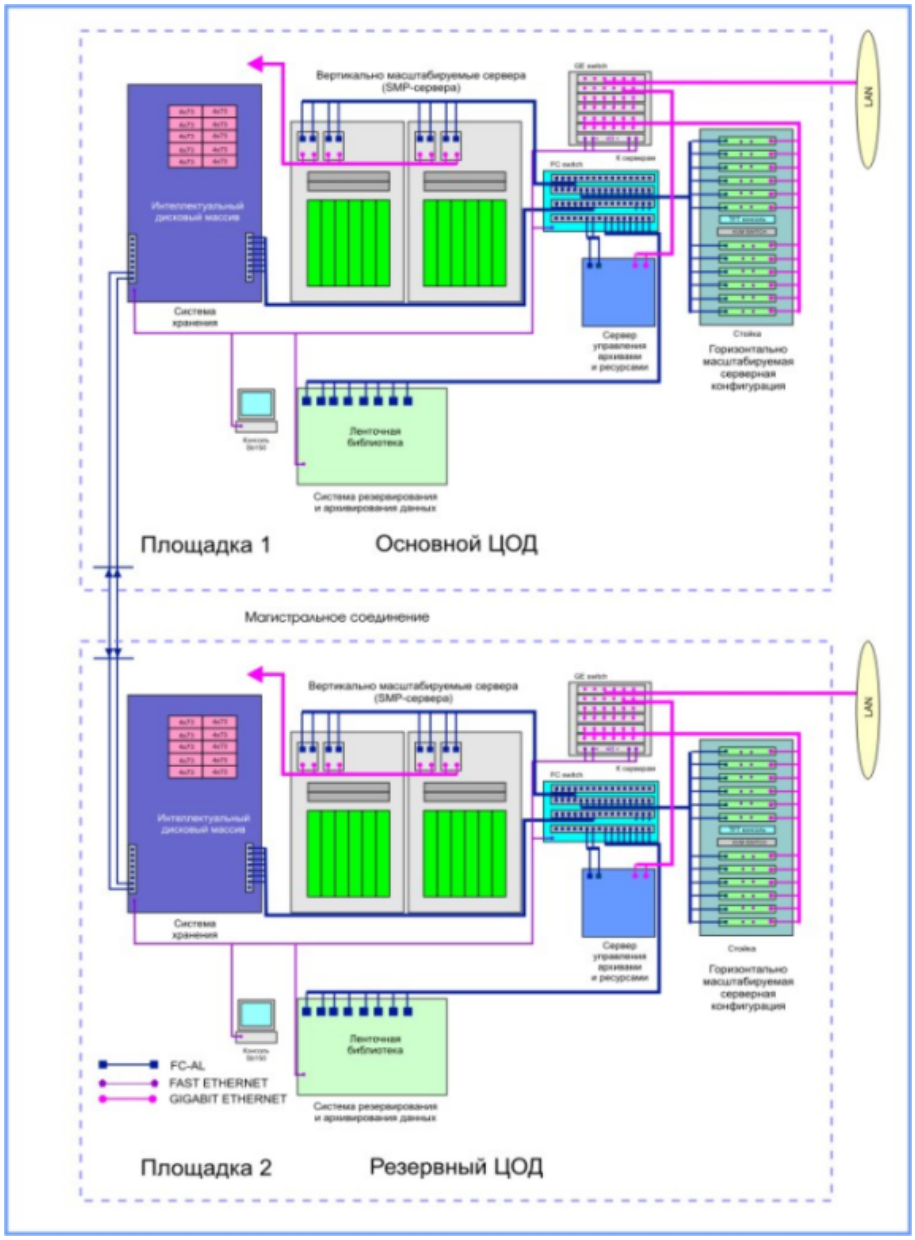

Рис. 3.1: Пример схемы организации ЦОД

4. Перевод на новую технологическую платформу существующих специализированных прикладных систем и АРМ должен осуществляться в рамках процесса модернизации прикладной программно-технологической инфраструктуры, при котором определяются приоритеты решаемых задач, пересматривается функциональность и обосновывается необходимость дальнейшего существования этих систем в виде отдельных или включение их состав вновь разрабатываемых интегрированных решений.

Состав компонентов БПТИ должен быть ориентирован на преимущественное использование ПО с открытым кодом и может варьироваться для конкретных программных и аппаратных решений, используемых при реализации прикладных систем. Это позволяет оптимизировать затраты на их внедрение и сопровождение, и повысить общую защищенность комплекса. Конкретные технические решения для реализации требуемой функциональности могут выбираться в зависимости от уже использующихся программных и аппаратных средств, квалификации персонала и текущего состояния развития технологий. Использование продуктов с открытым кодом дает широкий выбор таких решений, хотя не исключает возможность использования коммерческих и закрытых продуктов, работающих по открытым стандартам.

В перспективной архитектуре БПТИ выделяются следующие компоненты:

- операционные системы;

- пользовательские утилиты общего назначения;
- серверные компоненты, обеспечивающие сетевые сервисы общего назначения;
- средства администрирования программно-аппаратного комплекса;
- технологическая платформа информационных порталов, интеграции информационных ресурсов, быстрой разработки.

Схема перспективной архитектуры БПТИ приведена на рис. Рис. 3.2

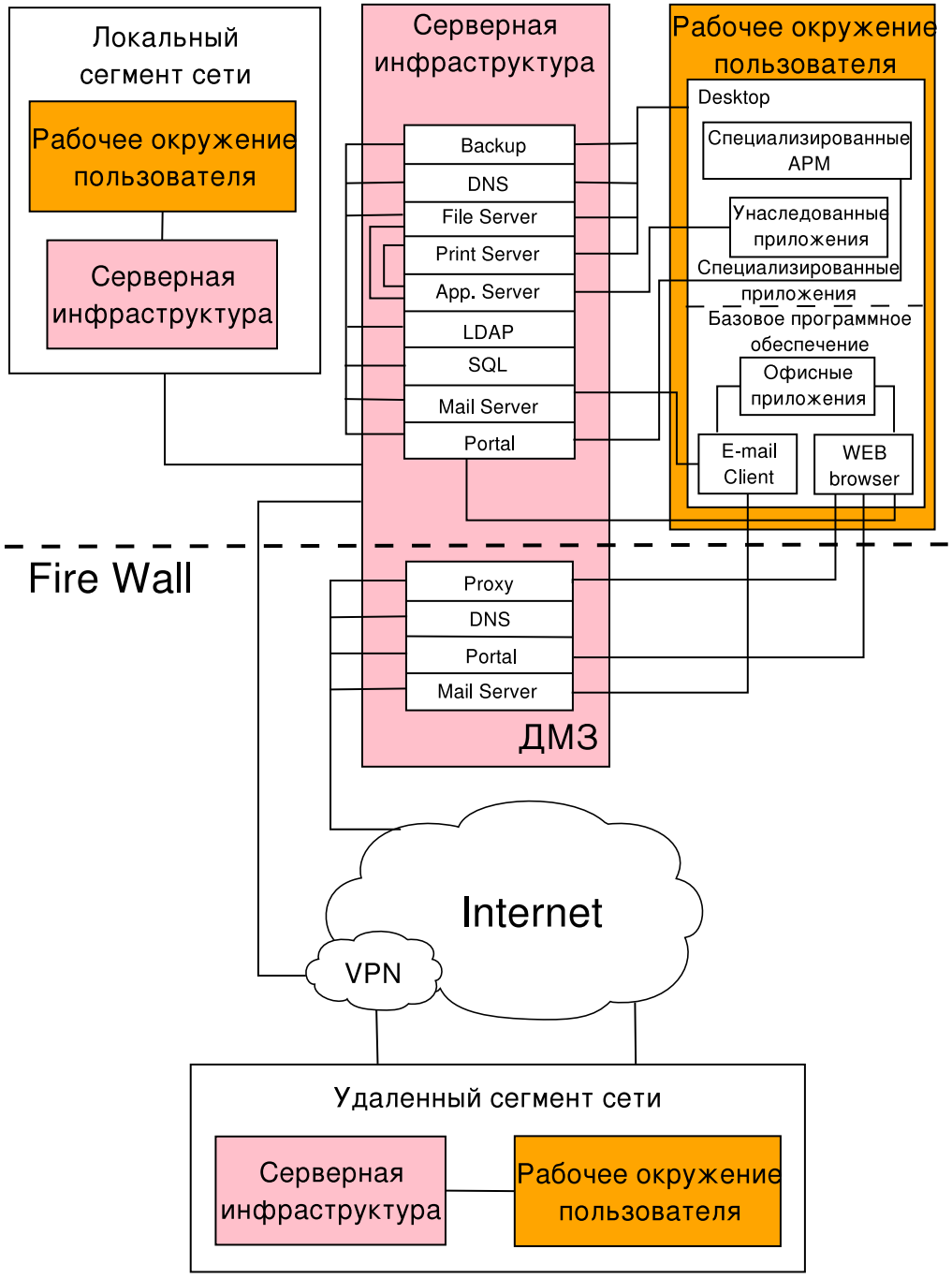

Рис. 3.2: Перспективная архитектура БПТИ

3.2.1 Операционные системы

Базовым компонентом программных комплексов является ядро операционной системы, обеспечивающее нижний уровень взаимодействия оборудования, программных компонент и пользователей. Ядро операционной системы Linux поддерживает различные аппаратные архитектуры: Intel x32, AMD/EMT 64, Itanium, ARM, PPC и др. Система драйверов поддерживает устройства PCI, USB, PCMCIA, SCSI, ISA и других наиболее распространенных стандартов. Поддерживаемые файловые системы: EXT2/3, FAT, NTFS, UFS, ReiserFS, XFS, JFS и др. Поддерживаемые сетевые файловые протоколы: IPv4, IPv6, IPX/SPX, AppleTalk, DecNet. Поддерживаемые сетевые файловые системы: NFS, SMB (Windows), Novell, Coda, AFS и др. Для построения систем повышенной защищенности могут использоваться механизмы CryptoAPI и SELinux.

Для Linux существуют развитые среды виртуализации, как свободные (UML, QEMU, Xen), так и коммерческие (VMware). Их использование позволяет повысить общую надежность комплекса, защищенность и более эффективно распределять вычислительные ресурсы.

3.2.2 Утилиты системного и общего назначения

Стандартом де-факто в качестве базового набора системных утилит, отвечающих промышленному стандарту POSIX, является набор программ проекта GNU, используемых практически во всех свободных UNIX-подобных системах.

В качестве графической подсистемы используется система X-Window. Она представляет из себя модульную систему, базовая часть которой осуществляет взаимодействие с устройствами ввода (клавиатура, мышь, сенсорный экран и др.) и обеспечивает отображение графических примитивов на дисплее. Эти функции могут выполняться как на локальных рабочих местах, так и по сети на удаленных рабочих местах. Это позволяет создавать терминальные решения, работающие по схеме «тонкий клиент», используя при необходимости устаревшее оборудование. Дополнительные расширения позволяют осуществлять непосредственный вывод в видео-память, ускоряют работу с трехмерными примитивами и т.д.

Рекомендуется проводить замену имеющихся закрытых компонент на функциональные аналоги с открытыми исходными текстами и работающие в среде Linux. При выборе данных компонент следует руководствоваться критериями доступности исходного кода, функциональной полноты, эргономичности, распространенности, наличием устойчивого сообщества разработчиков, обеспечивающих развитие проекта.

В качестве возможных вариантов замены, удовлетворяющих перечисленным критериям, можно привести графические пользовательские интерфейсы Gnome, KDE, а также продукты, приведенные в Табл. 3.1.

Таблица 3.1: Функциональные аналоги

Windows приложение	Функциональный аналог
Microsoft Office	OpenOffice.org
Internet Explorer	Mozilla, Firefox
Microsoft Visio	Dia
Adobe Photoshop	The Gimp
Lotus Notes (как почтовый пакет)	Evolution

Программное обеспечение, не имеющее функциональных аналогов, или необходимое для работы с ранее созданным информационным наполнением, работа с которым под Linux невозможна (Microstation, MGE, Lotus Notes, Corel Draw, AutoCAD и т.п.), можно запускать в режиме эмуляции Windows (Wine) или DOS (Dosemu), что не требует лицензионных отчислений на организацию сред исполнения унаследованных приложений. Здесь необходимо тщательное тестирование работоспособности приложения в среде эмуляции и гарантии того, что в приложения более не будут вноситься изменения, в противном случае приложения могут выполняться удаленно с использованием технологии Windows Terminal Server.

3.2.3 Серверная инфраструктура

- Веб-сервер предназначен для:
  - обслуживания статических компонент портала;

  - промежуточной буферизации наиболее часто запрашиваемых динамических страниц;
  - создания защищенных соединений по механизму SSL/TLS;
  - организации нескольких виртуальных веб-сайтов на одном физическом хосте.
- Сервер каталогов (LDAP) предназначен для хранения:
  - учетных записей пользователей и групп, для авторизации в других модулях (РОП, портал и т.д.);
  - общих конфигурационных параметров комплекса;
  - различных вспомогательных каталогов для обслуживания прикладных систем (например, организационная структура, справочники контрагентов и т.д.), укладывающихся в иерархическую модель данных.
- Серверы SQL-БД предназначены для хранения данных прикладных систем, организованных с использованием реляционной модели (например, БД документов СЭД, данные ГИС, и т.д.). В качестве SQL-ориентированных реляционных СУБД могут использоваться как свободно распространяемые продукты с открытым кодом (Postgres, MySQL), так и имеющиеся в организации коммерческие продукты (Oracle).
- Файловый сервер предназначен для предоставления сервиса удаленной работы с файловой структурой:
  - NFS — для рабочих мест с ОС класса Unix и Linux;
  - SAMBA — для рабочих мест с ОС Windows.
- Сервер печати предназначен для предоставления сервиса удаленной печати:
  - CUPS — для рабочих мест с ОС класса Unix и Linux;
  - SAMBA — для рабочих мест с ОС Windows.
- Сервер имен предназначен для разрешения доменных имен Интернет (DNS).
- Серверы приложений служат для запуска на них Dos, Windows приложений с клиентских машин Windows (WTS), Unix или Linux.
- Backup-сервер предназначен для предоставления сервиса удаленного резервирования данных.
- Серверы удаленной загрузки (PXE, BOOTP, TFTP) предназначены для осуществления возможности сетевой загрузки бездисковых рабочих станций и терминалов.

3.2.4 Средства администрирования программно-аппаратного комплекса

Контроль, управление и мониторинг программно-аппаратного комплекса обеспечивается набором программных компонентов, развернутых в среде одного из серверов. Организуется автоматический сбор и централизованное хранение диагностической информации с аппаратных и программных компонентов с использованием технологии SNMP (Simple Network Management Protocol), Syslog. Мониторинг состояния и управление комплексом осуществляется через стандартные механизмы удаленного доступа, такие как SSH, X-Window, VNC или через специализированные веб-интерфейсы.

3.2.5 Технологическая платформа интеграции, быстрой разработки приложений

В качестве технологической платформы интеграции представляется информационный портал, который должен обеспечить сопряжение рабочих мест с иными компонентами ИТ-инфраструктуры. Более подробное описание различных технологий, используемых при создании информационных порталов, приведено в Разд. Е.5.

3.2.6 Размещение компонентов БПТИ и организация удаленных сегментов

В структуре ЛВС выделяются две зоны для размещения компонентов БПТИ:

1. зона внешнего доступа из сети Интернет (ДМЗ). Предназначена для размещения внешних сервисов (WEB, E-mail, DNS и др.) в сети Интернет и обеспечения контролируемого доступа к ее ресурсам;

2. внутренняя закрытая зона ЛВС, изолированная от публичной сети Интернет и предназначенная для размещения серверов и рабочих мест сотрудников администрации. Все контакты с публичной сетью осуществляются только через специально разрешенные каналы для строго оговоренных сервисов и лиц.

Предоставление сервисов внешним контрагентам (гражданам, муниципальным образованиям и внешним организациям) осуществляется посредством публично-доступной сети Интернет с возможностью изоляции сетевого трафика в VPN.

При организации географически удаленных локальных сегментов БПТИ, связанных низкоскоростными каналами, компоненты серверной части БПТИ, предъявляющие высокие требования к пропускной способности телекоммуникационной среды, должны быть вынесены в эти локальные сегменты. К таким компонентам относятся:

1. файловый сервер (NFS, SAMBA);
2. сервер печати (CUPS, SAMBA);
3. сервер приложений (WTS, Linux);
4. сервер удаленной загрузки (PXE, BOOTP, TFTP);
5. backup-сервер.

3.3 Организация рабочего окружения пользователя (РОП)

РОП — это унифицированный набор технологий, программных и аппаратных средств, обеспечивающих повседневную работу пользователя, работу с различными автоматизированными системами и информационными ресурсами.

Комплекс программных средств РОП предназначен для обеспечения работы пользователей по следующим направлениям повседневной рабочей деятельности:

1. работа с офисными документами различных типов (текстовыми, электронными таблицами, презентациями и т.п.);
2. обеспечение доступа в интернет и интранет (браузер веб-страниц, электронная почта и т.п.);
3. запуск и работа с приложениями и ресурсами;
4. запуск и работа с унаследованными DOS и Windows приложениями;

РОП имеет в своем составе базовый набор программных средств, общий для работников всех функциональных подразделений, а также РОП включает дополнительный набор программных средств, позволяющий конфигурировать рабочие места для каждого конкретного исполнителя в соответствии с отведенной ему ролью (руководитель, администратор, исполнитель того или иного подразделения и т. д.).

Целями создания комплекса программных средств под названием «Рабочее окружение пользователя» являются:

1. унификация и стандартизация основного набора программного обеспечения, используемого в повседневной работе большинством сотрудников администрации;
2. приведение циркулирующих в администрации документов к единому формату;
3. единообразие в работе для всех сотрудников администрации;
4. снижение расходов на поддержание программного обеспечения и обучение сотрудников.

Предлагаемое решение позволяет использовать различные варианты модели вычислений, приведенных на Рис. 3.3. Благодаря использованию специализированных программных средств обеспечивается возможность доступа должностных лиц к автоматизированным системам, работающим на платформах семейства Microsoft Windows.

Вариант 1 описывает полноценное рабочее место, являющееся средой выполнения различных клиентских приложений. Технические требования, предъявляемые к такому рабочему месту, зависят от состава выполняемых на нем приложений и являются, как правило, высокими.

Варианты 2, 3 и 4 описывают разные технологии использования рабочего места в качестве «тонкого» клиента. Эти технологии позволяют существенно снизить технические требования к рабочим станциям за счет ограничения набора выполняемых на них приложений, также снизить требования к пропускной способности каналов связи за счет сокращения объемов передаваемых данных.

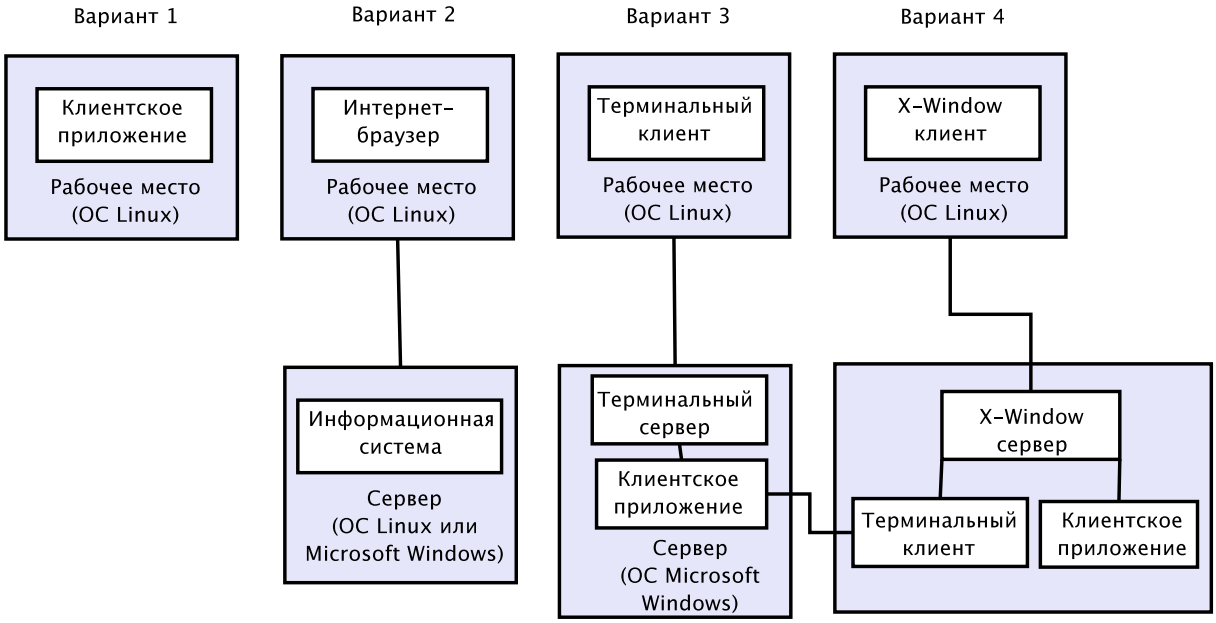

Рис. 3.3: Варианты РОП

3.4 Геоинформационная система (ГИС)

Геоинформационная система (ГИС) должна обеспечивать выполнение набора базовых функций, необходимых для работы с картами, кадастровыми планами и другими подобными объектами в электронном виде. В качестве основы ГИС предлагается использовать компоненты, построенные в соответствии с принципами программного обеспечения с открытым кодом (Open Source). На начальных этапах должен быть построен Опытный образец ГИС, демонстрирующий ее основные свойства. Состав и содержание Опытного образца ГИС будет определен на стадии предпроектного обследования и составления уточненного технического задания. Возможный вариант реализации Опытного образца ГИС:

1. интерфейс оператора ГИС — GRASS GIS;
2. система хранения — PostGIS;
3. модули для сопряжения геоинформационных данных с общим информационным пространством портала через веб-интерфейс — MapServer, WMS Adapter.

Общая схема построения системы приведена на Рис. 3.4.

3.5 Преимущества и особенности предлагаемых решений

Предлагаемые технологии и программные компоненты РОП обладают следующими преимуществами:

- все предлагаемые программные компоненты обладают нулевой стоимостью лицензий на использование;
- в составе предлагаемых решений имеются развитые средства управления (в том числе удаленного) программными компонентами, развернутыми на рабочих станциях, что позволяет администратору системы гибко и оперативно конфигурировать рабочие места в соответствии с функциональными потребностями оператора рабочего места;
- возможные варианты модели вычислений позволяют оптимально распределять вычислительную нагрузку между рабочими станциями и серверами, входящими в состав ЦОД;

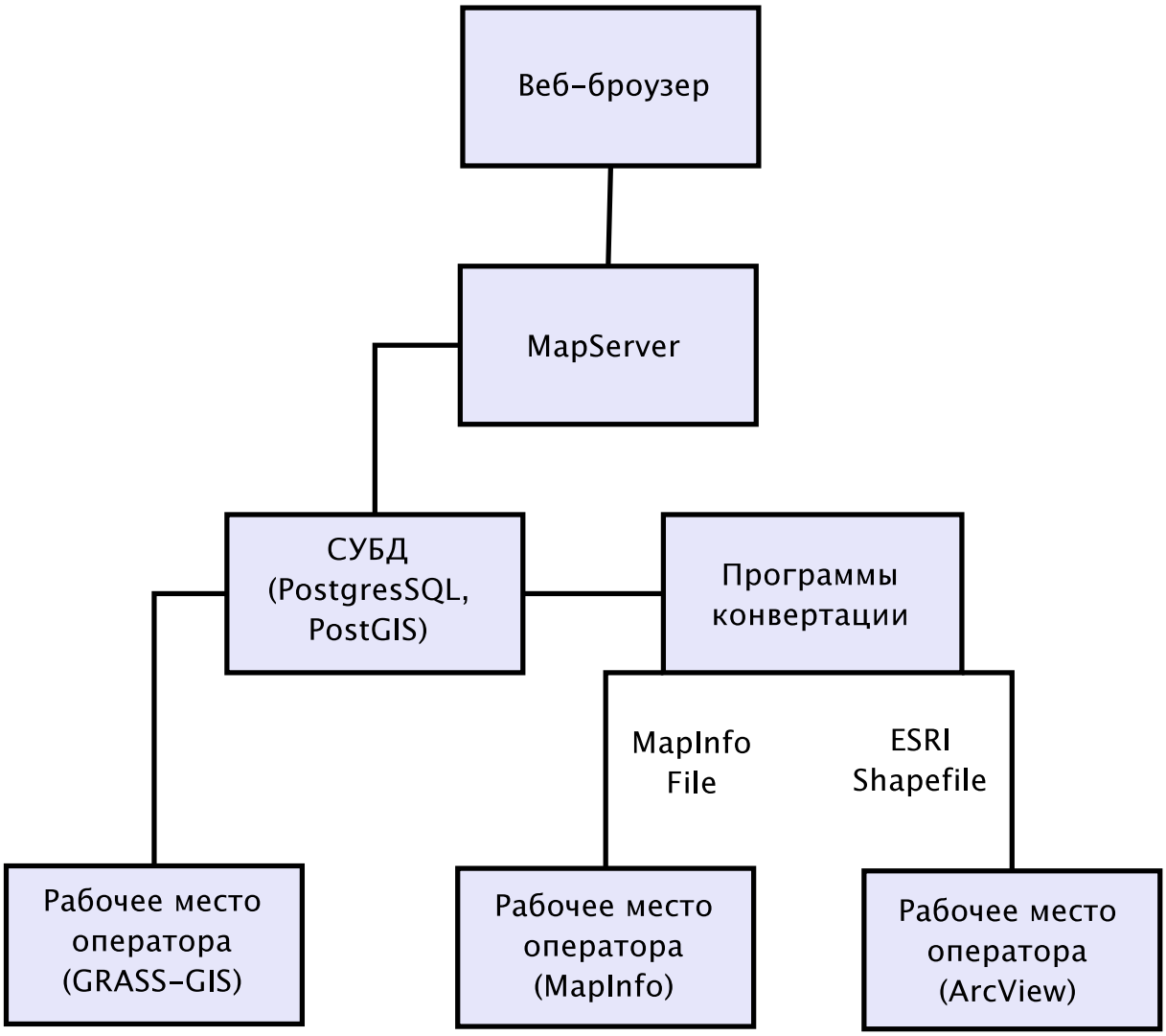

Рис. 3.4: ГИС

- применением модели вычислений «тонкий клиент» обеспечивается возможность дальнейшего использования морально устаревших компьютеров, которые уже не могут эксплуатироваться в качестве полноценных рабочих станций;
- используемая операционная система Linux обеспечивает высокую степень информационной безопасности и надежности;
- обслуживание рабочих станций, работающих под управлением операционной системы Linux, может выполняться удаленно, в том числе с использованием низкоскоростных каналов связи.

Для тиражирования и отчуждения от конкретного поставщика услуг решений и технологий, примененных в ходе построения РОП, будет создан специализированный типовой комплект программных компонентов, эксплуатационной и методической документации. Такой комплект может использоваться при дальнейшей модернизации РОП в любых учреждениях, органах власти и местного самоуправления.

Часть II

Выбор Linux

Глава 4

Миграция на Linux

В настоящий момент Linux занял свое место на рынке операционных систем для настольных компьютеров, постепенно он становится все более привлекательными для широкого круга конечных пользователей, особенно для тех, чья деятельность тесно связана с интернетом. Это вызвано, с одной стороны, теми ограничениями и проблемам, с которыми все в большей степени сталкиваются пользователи коммерческих настольных ОС. С другой стороны, дистрибутивы Linux становятся более мощными и удобными для конечных пользователей, что является естественным следствием усилий широкого сообщества программистов, разработчиков программного обеспечения с открытым кодом.

Это руководство можно использовать как технический справочник для ИТ-организации, планирующей использовать на своих настольных компьютерах операционную систему Linux. В сети доступно огромное количество текстов «HOW TO» для Linux, которые посвящены техническим деталям установки системы на различных платформах, настройке операционной системы, пользовательского интерфейса и другим вопросам.

В этом руководстве вы также найдете некоторое количество технических советов типа «HOW TO», но основной акцент сделан на более общих вопросах, мы ставим свой целью указать читателю на важные моменты, которые следует учесть и спланировать, составляя проект перехода на новую систему. Мы исходим из предположения, что миграция происходит от одной из операционных систем Microsoft Windows, и постараемся дать наиболее полное и последовательное описание технических решений и методов, необходимых для успешного перехода конечных пользователей на Linux-систему.

4.1 Как обстоит дело с миграцией сегодня

Даже за то время, пока писалось это руководство, произошло значительное расширение набора и возможностей программного обеспечения для Linux. Для тех немногих ИТ-организаций, которые сейчас стоят перед выбором базовой операционной системы для своих настольных машин, ситуация сравнительно проста, поскольку Linux предоставляет полный набор приложений, покрывающий все их нужды. Но зачастую речь идет не о новой установке, а о переходе на работу под Linux настольной вычислительной машины, работающей под управлением другой операционной системы. Для конечных пользователей наиболее важным вопросом будет вопрос перехода от набора программ, входящих в пакет Microsoft Office, к соответствующим приложениям, работающим под управлением Linux.

Кроме того, большинство пользователей не работают изолировано, а включены в сложную сетевую инфраструктуру, как минимум они пользуются средствами обмена электронными сообщениями (а Microsoft Outlook®, например, под Linux не работает). Наша книга содержит раздел, в котором обсуждаются и демонстрируются методы включения Linux-клиента в существующую сеть со службой каталогов Active Directory® и сервером Microsoft Exchange.

Успех миграции офисных приложений во многом зависит от того, насколько широко в организации используются продвинутые возможности Microsoft Office, а также собственные приложения, интегрированные или расширяющие возможности Microsoft Office. Наибольшие шансы успешно перейти на офисные приложения под Linux имеют пользователи, рабочие места которых соответствуют типу «Фиксированные функции» или «Техническая рабочая станция». Описание этих типов рабочих мест можно найти в Разд. 6.1, далее в Разд. 6.1.1 эти вопросы обсуждаются более подробно.

ВНИМАНИЕ

Сейчас и в ближайшее время только те пользователи, которые используют разумно-ограниченную функциональность офисных приложений, могут уверенно мигрировать под Linux. Так как офисные приложения под Linux постоянно совершенствуются (вместе с инструментами для соответствующих преобразований данных), то в скором времени можно ожидать, что это ограничение будет снято и продвинутые пользователи также смогут перейти на Linux.

4.2 Поиск оптимального решения

Организации, которые пользуются ПО с открытым исходным кодом для обеспечения работы серверов, например, большие банки, страховые компании, государственные учреждения и предприятия розничной торговли, естественно склоняются к тому, чтобы установить Linux также на настольных машинах своего персонала. Однако никакое, даже самое хорошее решение, не может быть одинаково пригодно для всех. Прежде чем планировать миграцию, ответьте на три критических вопроса:

- Существенно ли деятельность персонала зависит от программ независимых производителей, от плагинов или аппаратных средств, которые работают только под управлением Windows?
- Разработаны ли для рабочих станций персонала собственные приложения, которые основаны на использовании Win32® API и/или скриптовых языков Windows, например, Visual Basic®?
- В большой ли степени персоналу требуется в работе совместимость документов (это могут быть шаблоны или форматы файлов) с Microsoft Office?

Если ответ хотя бы на один из этих вопросов «да», то переход на Linux, возможно, не лучшее решение, во всяком случае, переход не будет простым и будет требовать известных вложений.

Если вы ответили «нет» на все три вопроса, то это означает, что ситуация благоприятна для перехода на Linux, и вам следует начать планировать пилотный вариант миграции (используя эту книгу как руководство). На основе пилотной миграции вы сможете сделать окончательные выводы о том, стоит ли переходить к промышленной эксплуатации.

4.3 Стратегическое планирование

Переход на Linux представляет собой только часть проблемы. Рано или поздно всякая организация сталкивается с необходимостью оптимизации и упрощения ИТ-инфраструктуры, включающей серверы, базы данных, приложения, внутренние сети, программное обеспечение рабочих станций и т.д. В противном случае она не будет иметь базы для развития и применения новых технологий.

При этом обсуждать и рассматривать следует в первую очередь функциональность того или иного элемента, а не отдельные продукты и бренды. Нет ничего хорошего в том, что мы помним название инструмента и теряем название функции, которую он выполняет. Например, словосочетание «текстовый процессор» почти никому не известно, а «Microsoft Word» слышал почти каждый. Между тем, Microsoft Word всего лишь один из текстовых процессоров, работающий на платформе Windows и Apple Macintosh, а есть и другие платформы и для каждой из них существует известный выбор текстовых процессоров.

Очень редко настольный компьютер работает сам по себе. Как правило, он подключен к серверам, банкам данных, принтерам. Кроме операционной системы на нем установлен известный набор приложений, инструментальных средств, зачастую — специально разработанное программное обеспечение. Всем этим надо управлять, все должно поддерживаться в рабочем порядке и сопровождаться.

Эта книга посвящена главным образом проблемам перевода на Linux клиентских машин, работающих на платформе Microsoft Windows. Мы не ставим себе задачу описать полностью стратегию миграции на Linux всех

элементов ИТ-инфраструктуры, однако будем иметь в виду, что миграция рабочей станции есть часть общего плана. Рис. 4.1 [1] в форме диаграммы демонстрирует пример полной миграции ИТ-инфраструктуры на платформе Windows NT® к инфраструктуре на базе Linux. Пунктиром выделена та часть миграционного плана, которая будет в центре внимания этой книги.

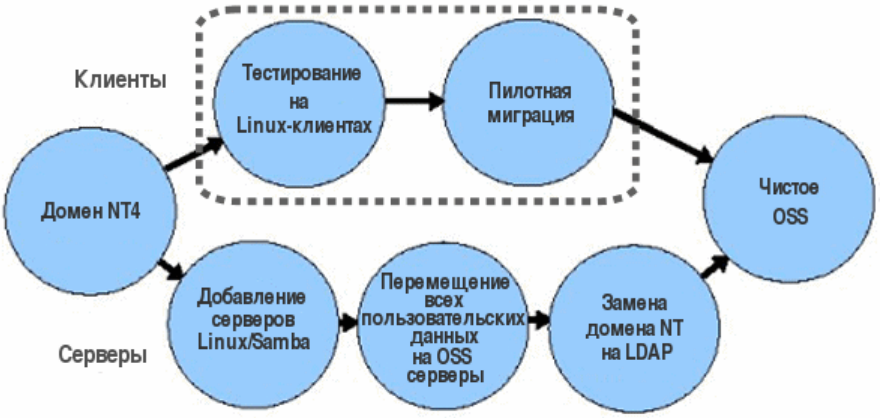

Рис. 4.1: Миграционный цикл

4.4 Планирование на уровне клиентов

Выбор платформы для конечного пользователя зависит от тех функций, которые он выполняет, и от тех приложений, которые он использует в своей работе.

Все больше бизнес-приложений разрабатываются с использованием кросс-платформенных средств, не зависящих от операционной системы, использующих открытые стандарты и широко распространенные технологии, такие как веб-броузеры. Наблюдается повсеместное движение в сторону сервис-ориентированной архитектуры (Service-Oriented Architecture, SOA), которая позволяет составлять приложение из уже существующих сервисов. Такой подход упрощает обмен данными и дает выигрыш за счет использования готовых компонент и сходных проектных решений внутри всего предприятия.

SOA часто реализуется посредством веб-сервисов. Веб-сервис характеризуются набором стандартных свойств, что гарантирует их способность к взаимодействию с другими приложениями, они используют такие общие технологии как XML (Extensible Markup Language), SOAP (Simple Object Access Protocol), HTTP (Hypertext Transfer Protocol) и другие.

Клиенты для приложений, основанных на веб-сервисах, часто написаны на Java или используют средства доступа веб-броузера. При таком подходе для работы приложения не важно, какая операционная система установлена на клиентской машине, что увеличивает гибкость системы в целом и расширяет возможность выбора. Критериями выбора приложения становятся его цена, надежность, гибкость, возможность поддержки и сопровождения, способность к взаимодействию с другими приложениями и другие.

Однако для многих предприятий такая технология, несмотря на всю ее привлекательность, может быть только целью, к которой они начинают движение, поскольку заменить установленное и работающее программное обеспечение так же сложно, как модернизировать старую технику.

Таким образом, принимая решения об обновлении технологии и архитектуры ИТ-системы предприятия, необходимо в полной мере учитывать требования и нужды пользователей на каждом рабочем месте. Глядя вперед, не стоит забывать и о сегодняшнем дне, принятые решения должны не только помочь внедрению новых прогрессивных подходов, но и обеспечить существующую сегодня функциональность.

[1] Источник: *The IDA Open Source Migration Guidelines*, netproject Ltd © European Communities 2003

4.5 Чем хорош Linux

Сегодня мы можем видеть, что Linux развился в мощную операционную систему, которая может работать практически на любом существующем оборудовании. Как правило, Linux требует меньше памяти и мощности процессора, чем другие операционные системы, и обеспечивает большую производительность.

Благодаря своему дизайну и открытому характеру программного обеспечения, Linux легко настроить под любые нужды. Linux распространяется на условиях лицензии GPL (GNU General Public License[2]) и может быть получен бесплатно. Однако, многие предприятия покупают готовые дистрибутивы Linux, поскольку это упрощает установку и обеспечивает поддержку со стороны производителя дистрибутива. Ключевым фактором выбора Linux является его открытость и гибкость, а не низкая цена дистрибутива. Широкая функциональность, надежность, масштабируемость, наличие поддержки — вот что обеспечило движение Linux из чисто академической сферы в область промышленного использования.

Благодаря поддержке ряда крупных компаний, таких как IBM, например, под Linux теперь работают многие популярные пакеты, среди которых стоит упомянуть Lotus Notes®, веб-броузер Mozilla, офисные приложения с открытым кодом, Java desktop и многие другие. Таким образом, Linux обеспечивает полный набор функциональности стандартного настольного компьютера.

Linux поддерживает стандарт POSIX (Portable Operating System Interface), определяющей такие детали работы UNIX®-подобных операционных систем, как формат системных вызовов и интерфейсов. Совместимость с POSIX позволила разработчикам портировать под Linux многие популярные приложения и утилиты UNIX.

Linux также включает полную реализацию сетевого пакета TCP/IP. Поддерживается весь набор клиентов и сервисов, включая стандартный интерфейс сокетов, благодаря чему все программы, которые используют TCP/IP, могут быть легко перенесены под Linux.

Linux поддерживает файловую систему для CD-ROM стандарта ISO-9660, программы управления печатью, мультимедийными устройствами, модемами и т.д. Коротко говоря, Linux предоставляет средства для обеспечения всех потребностей широкого круга конечных пользователей.

4.6 Обзор Linux и вариантов дистрибутивов

В 1984 году Фондом Свободного Программного Обеспечения (Free Software Foundation, FSF), некоммерческой организацией, созданной Ричардом Столлманом, была начата работа над проектом GNU, свободной версии операционной системы UNIX. Разработчики ставили своей целью создать не просто бесплатную версию операционной системы, но такую, исходные коды которой будут всегда доступны для чтения, модификации и распространения. К 1990 году в рамках проекта GNU были разработаны многие компоненты, включая компиляторы и текстовый редактор, однако свободного UNIX-подобного ядра не существовало. В 1991 году финский студент Линус Торвальдс начал разрабатывать такое ядро. При этом с самого начала работы исходные тексты новой операционной системы свободно распространялись через интернет под лицензией GPL, благодаря чему в разработку и тестирование включилось большое число программистов и работа пошла быстрее. В 1992 году была выпущена первая законченная версия ядра операционной системы, названной Linux.

Для того чтобы получить работающую операционную систему, ядра недостаточно. С самого начала в операционную систему входили утилиты GNU и другие программы. Такой набор необходимых для работы компонент стали называть дистрибутивами. Среди наиболее популярных дистрибутивов можно назвать Fedora и Red Hat Enterprise Linux, openSUSE и SUSE Linux Enterprise, Debian и производный от Debian Ubuntu Linux[3].

Linux — UNIX-подобная операционная система, соответствующая стандартам POSIX, распространяемая под лицензией GNU. То есть, она может быть получена бесплатно. Linux поддерживает несколько систем управления оконным интерфейсом и все возможные утилиты интернета, такие как FTP (File Transfer Protocol), Telnet, SLIP (Serial Line Internet Protocol). Он работает на 32-х и 64-разрядных процессорах, поддерживает многозадачность и работу с виртуальной памятью, разделяемыми библиотеками, сетевым протоколом TCP/IP. Linux обращается к собственной библиотеке обработки потоков (нитей) POSIX, что обеспечивает высокопроизводительную многопоточную обработку; симметричную мультипроцессорную работу (SMP), до 16 логических

[2] Полный текст лицензии GPL можно найти на странице: http://www.gnu.org/licences/licenses.html.
Проект GNU поддерживается Фондом Свободного Программного Обеспечения (Free Software Foundation, FSF). См.: http://www.gnu.org/fsf/fsf.html

[3] В России также известны русифицированные дистрибутивы ALTLinux, Linux Cyrillic Edition, ASPLinux (прим. переводчика)

CPU (или до восьми CPU-пар), а также для компьютеров Cray разрабатывается поддержка массированной мультипроцессорной системы MPP, которая может включать до 10000 процессоров AMD Opteron.

Linux может работать на архитектуре x86, Itanium®, AMD64, на серверах IBM серии eServer: zSeries®, iSeries™, pSeries®, на S/390®. Во всех случаях используется общий исходный код.

4.7 Будущее настольных систем под Linux

Мы наблюдаем сейчас динамичное развитие настольных систем под Linux и будущее выглядит многообещающим. Фирма Novell только что выпустила в свет новую версию промышленного настольного дистрибутива SUSE Enterprise Linux Desktop 10, а фирма Red Hat готовит к выпуску Red Hat Enterprise Linux 5, в который будет включено множество полезных нововведений, позаимствованных из открытого проекта Fedora. Обе фирмы, и Red Hat и Novell, производят свои дистрибутивы, ориентируясь на промышленное применение на различных предприятиях, от мелкого до среднего и крупного бизнеса. В то же время возрастает популярность дистрибутива Ubuntu Linux, ориентированного на применение в качестве настольной системы для домашних компьютеров.

В настоящее время перед сообществом разработчиков программного обеспечения для настольных систем под Linux в полный рост встал вопрос совместимости интерфейса различных программных продуктов, особенно это касается работы приложений в графическом окружении. Например, есть популярный мультимедиа плеер под Linux, но для того чтобы обеспечить его полную функциональность как для GNOME, так и для KDE, приходится затрачивать немало усилий. Причем разные компании решают эту задачу по-разному.

Для того чтобы сократить непроизводительны расходы сообщества по адаптации программного обеспечения, необходимо разработать стандарты, которые бы позволил однотипно включать приложения в различные дистрибутивы Linux. Эти стандарты должны поддерживаться производителями дистрибутивов Linux. Как раз сейчас организовалось широкое сообщество и ведется большая совместная работа, результатом которой станет дальнейшее развитие и распространение таких стандартов.

Важные разработки в этой области ведутся компанией Desktop Linux Working Group, спонсируемой OSDL (Open Source Development Labs). Проекты унификации взаимодействия приложений и различных видов рабочего окружения поддерживаются freedesktop.org. Более подробную информацию об этом можно найти по адресам:

- ODSL Desktop Linux Working Group:

  http://www.osdl.org/lab_activities/desktop_linux

- freedesktop.org:

  http://freedesktop.org/wiki/Home

Практически все ссылки на самую свежую информацию по этой теме можно найти на странице Free Standards Group (подробнее об этой группе см. http://www.freestandards.org/en/About). Одним из ключевых проектов, поддерживаемых этой группой, является Linux Standard Base (LSB), именно этот проект и то, какие приложения он имеет в настольных системах, будет обсуждаться далее в этой главе.

ЗАМЕЧАНИЕ

Далее в этой главе мы воспроизводим содержание следующего документа:

From Server to Desktop: Linux Standard Base Engages Application Developers Worldwide

Использование этого документа в нашем издании стало возможно благодаря разрешению, полученному от корпорации Intel и Free Standards Group.

Более подробную информацию о них можно найти по адресам:

http://www.intel.com/software/opensource

http://www.freestandards.org

4.7.1 От серверов к настольным системам: Linux Standard Base привлекает разработчиков программного обеспечения по всему миру

Возможности LSB в качестве стабилизирующей силы, способной сделать Linux еще более привлекательным для промышленного применения, особенно в настольных системах, не вызывают сомнений.

В течение многих лет сообщество разработчиков ПО с открытым кодом стояло перед задачей создания стабильной, унифицированной хорошо сопровождаемой операционной системы — надежной платформы для разработки пользовательских приложений для промышленного применения. В то время как с Linux уже завоевал популярность в качестве надежной, эффективной и недорогой ОС для промышленных серверов различного уровня, применение Linux на настольных компьютерах оставалось ограниченным. Главной причиной этого был недостаток полноценных офисных приложений, соответствующих требованиям, предъявляемым к программному обеспечению в большой организации. Созданию таких приложений во многом мешало отсутствие стандартизации графического рабочего окружения в различных дистрибутивах Linux.

Корпорация Intel и группа Free Standards Group (FSG), вместе с другими членами сообщества поставили перед собой цель создать новые возможности для разработчиков пользовательского ПО. Те препятствия, которые возникали при промышленном использовании UNIX и которые были следствием фрагментации и отсутствия единства, должны послужить уроком. Чтобы в будущем не повторялись прошлые ошибки, многие из тех, кто был вовлечен в эволюцию UNIX, объединились с целью разработки стандартов Linux, которые обеспечили бы его развитие и широкое применение.

Позитивным шагом в направлении стандартизации является создание LSB (Linux Standard Base), сейчас уже широко принятого, и утвержденного Международной организацией по стандартам (ISO) и Международной электротехнической комиссией (IEC). Официальное признание этого стандарта, безусловно, приведет к расширению круга пользовательских приложений для настольных систем, ускорению внедрения новинок и дальнейшему улучшению положения Linux на рынке ПО. Как независимые производители программного обеспечения, так и конечные пользователи выигрывают от того, что рынок ОС для настольных компьютеров становится более широким и гибким. Текущий момент является очень подходящим для компаний и разработчиков программно обеспечения, которые могут включиться в развитие и распространение LSB и ПО на его базе. Одним из таких проектов стандартизации является Linux Desktop Project, который был начат в 2005 и главной целью ставит стандартизацию графического рабочего окружения для Linux. Далее в этом разделе мы будем обсуждать этот проект подробнее.

4.7.2 Рабочий стол под Linux сегодня

Необходимость адаптировать свои программы для различных вариантов операционных систем сейчас ложится тяжелым грузом на плечи разработчиков пользовательских приложений. Когда речь шла о разработке программного обеспечения для Linux, то приходилось учитывать еще и различия дистрибутивов. Поэтому стандартизация для Linux стала жизненно необходима. Free Standards Group была создана с целью собрать все лучшее, что есть в этой области и создать стандарт, который касается как вопросов переносимости бинарного кода, так и концепций разработки пользовательского ПО. После утверждения международными организациями стандарта LSB появилась реальная возможность стандартизации приложений для рабочего стола под Linux.

Сейчас доля Linux на рынке программного обеспечения для настольных компьютеров не так велика, но если обратить внимание на тенденции, то становится ясно, что у Linux хорошие перспективы в качестве ОС для настольных систем.

Свежие дистрибутивы Linux имеют дружественный пользовательский интерфейс и обладают хорошим набором офисных приложений.

Как говорит исполнительный директор FSG Джим Землин (Jim Zemlin), по своей сути LSB является набором бинарных программных интерфейсов, программных библиотек и стандартов, обеспечивающих возможность взаимодействия. Этот набор позволяет разработчикам программного обеспечения не заботиться об особенностях того или иного дистрибутива, а рассматривать Linux-среду в качестве единого целевого окружения.

4.7.3 Состав Linux Standards Base

LSB построен во многом на базе уже существовавших ранее стандартов, кроме того, он описывает бинарный интерфейс между приложениями и средой исполнения. Стандарты, на базе которых был разработан LSB,

включают в себя Single UNIX Specification (SUS), System V Interface Definition (SVID), C++ ABI и System V Application Binary Interface (ABI). Кроме того, при построении LSB использовались стандарты POSIX на программный код и определения интерфейсов из SUS.

Разработчики LSB не только включили в состав стандарта удачные аспекты стандартов-предшественников, но учли ошибки и ограничения, в них допущенные. Например, POSIX определяет только программный интерфейс и не может гарантировать совместимость двоичного кода. На другом конце спектра находится стандарт OSF/1, который обеспечивает такую совместимость, но устанавливает слишком жесткие рамки. При разработке LSB была сделана попытка найти нечто среднее меду двумя этими подходами.

LSB устанавливает границы доступности интерфейсов внутри конкретных библиотек и содержит спецификацию структур данных и ограничений, связанных с каждым интерфейсом. Ниже на Рис. 4.2 можно видеть организацию компонент в LSB 3.1. Эти компоненты включают библиотеки, необходимые для разработки (например, C++), структуры файловой системы (определение мест расположения общих файлов), спецификации на поведение общедоступных интерфейсов, детали организации пакетов приложений, процедуры, выполняемые перед и после установки пакета и т.д.

Рис. 4.2: Организация компонент в LSB 3.1

Кроме LSB той же группой разработан LSB Software Development Kit (LSB SDK), который состоит из рабочего окружения и необходимых инструментов, позволяющих модернизировать приложения в соответствии с LSB. Инструменты, входящие в LSB SDK, позволяют разработчикам тестировать бинарные файлы и RPM-пакеты на соответствие стандартам LSB и контролировать использование API на этапе построения приложений.

Начальные версии стандарта LSB были сфокусированы на особенностях Linux-приложений для серверов, такой подход был оправдан существовавшей практикой. Версии, появившиеся в 2005 и 2006 годах дополнены спецификациями для приложений и библиотек Linux, ориентированных на настольные системы.

LSB версии 3.0 и версии 3.1, в большей степени ориентированной на рабочие станции, подготовили почву для появления LSB версии 4.0, которая будет содержать необходимую формализацию для разрешения многих известных моментов, создающих сейчас проблемы для разработчиков ПО под Linux. Одна из целей создания новой версии — дать возможность разрабатывать графические приложения для всех LSB-совместимых дистрибутивов, которые будут одинаково хорошо работать как в рабочей среде KDE, так и в рабочей среде GNOME. Подробную информацию о спецификациях LSB можно найти по адресу:

http://lsb.freestandards.org

4.7.4 Стимулы для независимых производителей программных продуктов

Природа разработок с открытым кодом такова, что вокруг новых приложений и дистрибутивов возникают новые модели деловых взаимоотношений. Внутри открытой и гибкой модели решения нового поколения могут заимствовать лучшие черты предшествующих разработок в комбинации с инновационными технологиями. Как следствие расширяется сфера рыночного применения приложений, в том числе в областях, которые раньше не были затронуты.

Ранее разработка ПО для настольных систем под Linux была затруднена отсутствием унификации. Вот как говорит об этом Ян Мердок, технический директор FSG, один из тех, кто был вовлечен в разработку стандартов UNIX в начале 90-х годов: «Раньше разработчики вынуждены были сидеть и выбирать, для какого дистрибутива они будут работать — это может быть Red Hat, или Novell, или Debian — и затем создавать свое приложение для каждого дистрибутива. Особенно это касалось рабочего стола. Не было стандартного пути для самых базовых вещей, таких как создание иконок на рабочем столе или интерфейс для общения с системой печати. Приходилось все время иметь в виду то окружение, в котором будет работать приложение. Не только выбор дистрибутива имел значение, но и то, какой менеджер рабочего стола будет запущен. Например, способы добавления пунктов меню в GNOME и KDE были разными, даже если использовалась одна и та же версия одного и того же дистрибутива. Позиция разработчиков ПО была такова: они хотели разрабатывать приложения, ориентированные на определенную платформу, и выбирали такую платформу, которая давала им наибольший возможный рынок.»

> «LSB обеспечивает разработчикам приложений под Linux единую платформу. Приложение, удовлетворяющее стандарту LSB, будет одинаково хорошо работать с любым дистрибутивом Linux, совместимым с LSB. Таким образом возможный рынок для такого ПО достигает максимума.»
>
> —Ян Мердок, технический директор Free Standards Group

Сейчас рынок программного обеспечения не ограничен географическими рамками, но в разных странах обычно популярны различные дистрибутивы. Так, в США распространены дистрибутивы фирмы Red Hat, в Европе — Novell, а в азиатских странах — свои дистрибутивы. Если разработчик ориентируется на определенный регион, то ему лучше выбрать соответствующий дистрибутив. Но это изначально накладывает ограничение на распространение ПО.

До версии 3.1 LSB в первую очередь был нацелен на стандартизацию компонент ядра, таких как среды исполнения C и C++ и других, ассоциированных с приложениями, работающими на серверах.

После того как LSB был утвержден в качестве ISO-стандарта, вышла версия 3.1, которая затрагивает нижний уровень приложений, работающих на настольных компьютерах, например, популярные инструменты разработки графических пользовательских интерфейсов GTK и Qt*.

Версия 4.0 призвана продолжить стандартизацию до уровня разработок для рабочего стола. Далее в этом разделе перспективы стандартизации, связанные с этой версией, обсуждаются более подробно.

4.7.5 Существенное увеличение числа потенциальных пользователей

Новые модели разработки позволяют коммерческим компаниям, разрабатывающим программное обеспечение, существенно увеличить круг потенциальных пользователей своей продукции. Во многих странах, таких как Бразилия, Китай и Индия, число пользователей персональных компьютеров постоянно увеличивается, значительное часть их работает под Linux. Вычислительные машины под Linux работают в образовании, государственных учреждениях, мелком и среднем бизнесе. Соответственно, потребность в пользовательском ПО для настольных машин существенно возрастает.

4.7.6 LSB-совместимые дистрибутивы

Разработчики дистрибутивов Linux могут вложить существенный вклад в дело повсеместного распространения стандартов. Выпуск LSB-совместимых дистрибутивов должен стать правилом для разработчиков, тестирование на соответствие стандартам должно быть включено на всех этапах разработки. Дистрибутивы ведущих производители уже сейчас достигли соответствия LSB-стандартам. Разработчики дистрибутивов должны принять участие в работе над стандартами.

4.7.7 LSB-совместимые приложения

Независимые производители ПО могут помочь расширению стандартизации, следуя стандартам LSB в своих разработках для ОС Linux. Они также могут поддерживать связь с FSG, если хотят, чтобы быть их концепции и требования были включены в следующие версии LSB.

4.7.8 Конечные пользователи

Конечные пользователи, выбирая операционную систему для своей машины или определяя политику при покупке программного обеспечения для своей фирмы, могут обратить внимание на соответствие продуктов LSB. Таким образом они выбирают наиболее прогрессивные технологии, получают более надежный продукт и более широкий выбор гарантированно совместимого программного обеспечения.

Исключительно полезным ресурсом по этой тематике является freedesktop.org, см.:

http://www.freedesktop.org

freedesktop.org — форум, посвященный разработке общих технологий для менеджеров рабочего стола системы X-Window, в первую очередь речь идет о KDE и GNOME.

4.7.9 Вехи на пути к широкомасштабному внедрению

Первой вехой можно считать утверждение LSB в качестве ISO-стандарта, которое произошло 1 ноября 2005 года на конференции Open Source Business Conference и было важным знаком публичного признания LSB и ОС Linux. Утвержденный ISO/IEC в качестве Публично-доступных Спецификаций (Publicly Available Specification, PAS), стандарт был опубликован как международный стандарт 23360 (International Standard 23360).

4.7.10 LSB Desktop Project

Следующей вехой мы считаем объявление о формировании проекта для настольных машин LSB Desktop Project, сделанное группой FSG 18 октября 2005 года. Проект был поддержан фирмой Intel и другими организациями. Сейчас уже среди участников этого проекта можно видеть такие заметные в компьютерном мире организации как Adobe, IBM, HP, Linspire, Mandriva, Novell, RealNetworks, Trolltech, Red Hat, Xandros и другие. Эти организации тратят время и свои ресурсы на работу по унификации общих библиотек и стандартизации поведения приложений в рамках LSB.

После того как будут окончательно стандартизованы требования на поведение приложений как во время выполнения, так и во время установки, разработчики пользовательских приложений для Linux будут освобождены от необходимости создавать библиотеки и версии своих программ, ориентированные на отдельные дистрибутивы, они смогут ориентироваться на Linux-платформу в целом.

До тех пор пока не вышла ожидаемая вскоре версия стандарта 4.0, разработчики могут опираться на требования LSB версии 3.1, поскольку эта версия представляет собой фундамент, на котором будет построена версия 4.0. По словам Яна Мердока, в промежутке между версиями будет иметь место некоторое развитие LSB, например, будет добавлена функциональность в наборе инструментов. Часть добавлений связана с предложениями и требованиями независимых производителей ПО. Изменения произойдут и в части, касающейся рабочего стола, там будут интегрированы наработки сообщества freedesktop.org.

LSB Desktop Project является часть общего проекта LSB и построен по тому же модульному принципу, это позволяет производителям дистрибутивов и ПО использовать стандарт в той степени, в которой это представляется для них возможным. Такой подход привлекает большое число промышленных лидеров и позволяет им включиться в работу по созданию действительно полезного и соответствующего требованиям как разработчиков, так и пользователей стандарта.

Политика Free Standards Group состоит в привлечении к активному сотрудничеству как можно более широкого круга специалистов, в том числе производителей дистрибутивов, независимых производителей ПО, сообществ разработчиков программ с открытым кодом и отдельных программистов. Разработка ясного и работоспособного стандарта существенно расширит базу применения ОС Linux в качестве операционной системы для настольных машин как на уже сложившемся рынке, так и на вновь возникающем.

Узнать больше о проекте Linux Desktop Project можно посетив страницу в интернете:

http://lsb.freestandards.org

Подписаться на лист рассылки и принять участие в формировании и уточнении стандартов для настольного компьютера под Linux можно по адресу:

http://lists.freestandards.org

Последние новости, касающиеся разработок для настольного компьютера под Linux, и дискуссии можно найти по адресу:

http://www.desktoplinux.com

Глава 5

Аргументы в пользу миграции

В этой главе мы будем обсуждать основные причины и обоснования необходимости миграции клиентских машин на Linux.

Мы не будем подсчитывать стоимость мероприятия или коэффициент окупаемости инвестиций, мы не собираемся продавать вам миграцию. Мы просто хотим указать на те причины, которые могут побудить вас перевести конечных пользователей на Linux.

Поскольку в этой книге описывается миграция от Windows-клиента к Linux-клиенту, то неизбежно возникает желание сравнить платформы. Однако здесь мы главным образом хотим обсудить аргументы в пользу миграции как таковой. Сравнения двух платформ постоянно появляются в различных статьях, записках и отчетах, масса ресурсов на эту тему присутствует в интернете; большинство из них так или иначе противоречат друг другу.

В этой главе вы найдете следующие разделы:

- Разд. 5.1 Зачем мигрировать

  Причины, по которым может возникнуть вопрос о миграции

- Разд. 5.2 Когда следует мигрировать, а когда нет

  Обсуждение факторов, влияющих на принятие решения о том, что миграция необходима

- Разд. 5.3 В чем заключаются особенности Linux?

- Разд. 5.4 Цели миграции

  Какие преимущества мы ожидаем получить в результате миграции

5.1 Зачем мигрировать

Прежде всего заметим, что желание заменить операционную систему на настольном компьютере конечного пользователя возникает тогда, когда появляется альтернатива, когда есть на что менять. Решение выполнить миграцию основывается на том, что альтернативный вариант по своим свойствам подходит больше, чем имеющийся. В этом разделе мы обсудим свойства, которыми обладает настольная машина, работающая под Linux.

С нашей точки зрения, главным образом на принятие решения влияют следущие аспекты:

- Безопасность персонального компьютера
- Полные затраты, связанные с ПО настольного компьютера
- Уровень управляемости клиента
- Удобство настройки клиента под нужды пользователя
- Философия открытого ПО
- Простота в эксплуатации

Рассмотрим каждое их этих свойств в деталях

5.1.1 Безопасность персонального компьютера

Linux имеет большое число различных средств, отвечающих за обеспечение безопасности, но сейчас мы сконцентрируем внимание на тех, которые тесно связаны с работой приложений на клиентской машине. Мы предполагаем, что клиент включен в сеть и на нем наряду с другими активно используются сетевые приложения. Это предположение определяет круг дальнейших тем обсуждения, а именно:

- Безопасность броузера
- Безопасность клиента электронных сообщений
- Защита пользователя и обеспечение безопасности
- Оперативность исправления ошибок
- Модульность операционной системы
- Брандмауэр

5.1.1.1 Безопасность броузера

Броузеры, используемые под Linux, как правило — программы с открытым кодом. Для некоторых из них существуют версии, работающие с другими операционными системами. Существенно то, что в Linux броузер никогда не интегрирован в операционную систему, это отдельное приложение, которое просто работает под управлением операционной системы. Наиболее часто используются следующие броузеры:

- Mozilla
- Opera
- Konqueror

Поскольку броузеры не интегрированы в операционную систему, при работе с ними практически невозможно разрушить систему безопасности операционной системы. Все броузеры могут запускаются в пространстве, выделенном для пользователей, и не могут затрагивать пространство, выделенное под ядро. Это также означает, что программное обеспечение броузера может быть установлено пользователем, не имеющим административных полномочий.

Конечно, ни одна программа не может быть полностью застрахована от слабых мест, но, во-первых прореха в коде броузера не дает пиратам возможности нарушить работу ядра, а, во-вторых, подобные проблемы под Linux очень быстро обнаруживаются и исправляются. Скорость реакции обусловлена тем, что с Linux связано широкое и активное сообщество разработчиков ПО с открытым кодом, и тем, что изменять и тестировать нужно только само приложение, а не операционную систему в целом.

До сих пор броузеры с открытым кодом имели дополнительное преимущество за счет того, что были менее распространены, чем Microsoft Internet Explorer®, и, соответственно, были не так привлекательны для пиратских атак. Однако, число установок свободных броузеров увеличивается, и это преимущество постепенно сходит на нет.

5.1.1.2 Безопасность клиента электронных сообщений

Под клиентом электронных сообщений мы будем понимать любую программу, которая применяется для обмена электронными сообщениями между двумя конечными пользователями с использованием протокола TCP/IP, включая интернет. Таким образом в понятие «клиент электронных сообщений» включаются как почтовые клиенты, так и клиенты для непосредственного обмена сообщениями.

Также как и броузеры, приложения для обмена электронными сообщениями под Linux — это программы с открытым кодом, которые не интегрированы в операционную систему, а просто работают под управлением операционной системы. Таким образом, все, что было сказано о безопасности использования броузеров под Linux, справедливо и для клиентов электронных сообщений.

Клиенты электронных сообщений с открытым кодом как правило могут работать с несколькими протоколами обмена сообщениями. Таким образом, выбор клиента под Linux не означает жесткой привязки к одному протоколу и оставляет возможность выбора как протокола, так и приложения, управляющего потоком сообщений

на стороне сервера. С другой стороны, выбор самих клиентов с открытым кодом довольно широк, что позволяет выбрать лучший вариант для конкретного приложения на серверной стороне.

5.1.1.3 Защита пользователя и обеспечение безопасности

Вопросы обеспечения безопасности системы и защиты конечного пользователя от несанкционированного доступа с самого начала были важной частью проекта по созданию операционной системы Unix. Linux, так же как и Unix, по своей сути является многопользовательской системой, в которой каждому пользователю и группе пользователей приписаны конкретные полномочия. В Linux только один пользователь, пользователь с полномочиями администратора, может производить существенные изменения в настройках операционной системы, что обеспечивает дополнительную безопасность как системы в целом, так и отдельных конечных пользователей.

Конечные пользователи под Linux, как правило, не имеют административных полномочий для доступа к операционной системе. Кроме того, конкретный пользователь или группа пользователей имеют различные права доступа к различным группам файлов. Благодаря этому данные каждого пользователя защищены от несанкционированного доступа.

5.1.1.4 Оперативность исправления ошибок

Никакая операционная система не может гарантировать отсутствие слабых мест в программном обеспечении. Поэтому очень существенным фактором безопасности является время реакции на информацию об ошибках. Риск, связанный с «брешью» в системе безопасности, прямо пропорционален времени, которое требуется, чтобы ее заделать. Поскольку большинство компонент в Linux — ПО с открытым кодом, всякая информация об ошибке вызывает реакцию большого числа программистов и, как правило, бывает исправлена очень быстро.

Существенным моментом является то, что каждая организация, работающая с открытым ПО, имеет доступ к исходным текстам. Поэтому всякий, кто обнаружил ошибку, может ее исправить, протестировать внесенные изменения и разослать соответствующий патч всем остальным пользователям. В случае же закрытого ПО нет ни возможности адекватной диагностики, ни возможности исправления. Единственное, что можно сделать — послать сообщение об ошибке производителю ПО и ждать, пока тот сам ее исправит.

5.1.1.5 Брандмауэр

В Linux сервисы брандмауэра встроены в ядро. Начиная с версии ядра 2.4, эти сервисы называются iptables. Правильно сконфигурированный iptables позволяет администратору определить набор правил (policy) для работы клиентов Linux в сети. Брандмауэр iptables может быть использован для:

- Защиты системы от проникновения вирусов
- Предотвращения использования запрещенных протоколов
- Блокирования известных средств взлома

5.1.2 Стоимость рабочего места под Linux

Очевидно, что часть затрат на организацию рабочего места с настольной машиной носят общий характер и не зависят от того, какая операционная система будет выбрана. Поэтому, в первую очередь, обратим внимание на те факторы, которые могут зависеть от операционной системы. Среди них:

- Стоимость лицензии и поддержки дистрибутива Linux
- Стоимость оборудования
- Стоимость базового набора приложений клиентской машины
- Затраты на сопровождение и поддержку
- Стоимость миграции

5.1.2.1 Стоимость лицензии и поддержки дистрибутива Linux

Ядро Linux и большинство приложений, включаемых в дистрибутив, являются свободным ПО и распространяются под лицензией GNU Public License (GPL). Это означает, что ПО может быть получено бесплатно. То есть, стоимость лицензии на рабочее место нулевая.

Однако промышленные дистрибутивы обычно стоят достаточно дорого. Дистрибутив представляет собой некоторый заранее подобранный комплект: ядро и набор приложений вместе с процедурой установки системы и документацией. Обычно фирмы, продающие дистрибутивы, включают в стоимость пакета поддержку на один год. Существуют различные формы поддержки пользователей, как правило, за дополнительную плату у тех же фирм можно получить поддержку более высокого уровня.

Поскольку Linux представляет собой свободное ПО, то его можно использовать полностью бесплатно, не тратя денег ни на лицензию, ни на поддержку. В этом случае организация может рассчитывать только на поддержку со стороны сообщества разработчиков свободного ПО, такой вариант оправдан при наличии в штате собственных программистов.

5.1.2.2 Стоимость оборудования

Большинство дистрибутивов Linux очень нетребовательны к оборудованию и хорошо работают на достаточно старых машинах. Вполне вероятно, что для установки клиентов Linux можно использовать те машины, от которых пришлось отказаться, поскольку они не соответствовали требованиям очередной версии Microsoft Windows.

Однако следует учитывать, что последние версии дистрибутивов Linux в варианте «промышленный» имеют довольно высокие требования к оперативной памяти, приближаясь по этому параметру к Microsoft Windows. Но, поскольку требуется нарастить только память, то, как правило, это требование легко выполнить даже для старых машин.

5.1.2.3 Стоимость базового набора приложений клиентской машины

Дистрибутивы Linux включают в себя большое число базовых приложений, как то: текстовые редакторы, средства обработки изображений, броузеры, средства для обмена электронными сообщениями, набор офисных приложений, включающий текстовый редактор, редактор электронных таблиц и средство подготовки презентаций. Таким образом, стоимость приложений для стандартного рабочего места практически равна нулю.

Даже если нужное приложение отсутствует в дистрибутиве, есть достаточно большая вероятность, что оно существует в виде свободного пакета, который можно бесплатно скачать из сети и установить.

5.1.2.4 Затраты на сопровождение и поддержку

Известно, что поддержание клиентской настольной машины в рабочем состоянии, в том числе профилактика и поддержка системы безопасности, требует определенных затрат. Это обстоятельство, безусловно, является веским аргументом в пользу миграции на Linux, так как Unix-подобные операционные системы включают массу опций, позволяющих минимизировать стоимость поддержки. Например, возможность удаленного доступа по протоколу telnet или ssh, позволяет устанавливать на клиентской машине нужные скрипты и удаленно выполнять их.

Используя удаленный запуск скриптов, несложно следить за порядком на клиентских машинах и делать необходимые поправки прямо с сервера. Например, если надо обновить версию пакета или установить заплатку на все клиентские машины, это можно сделать прямо с сервера при помощи скрипта, не прерывая работы конечных пользователей.

Существуют также и специальные инструменты для управления системой различных производителей, например, Red Hat Network фирмы Red Hat или ZENworks Suite фирмы Novell.

5.1.3 Уровень управляемости Linux клиента

Одним из наиболее важных вопросов при выборе платформы для клиентской машины является вопрос управляемости. Вопросы стоимости поддержания ПО на клиентской машине в рабочем состоянии обсуждались

в Разд. 5.1.3. Степень управляемости клиентов под Linux очень высока благодаря, во-первых, заложенным изначально в Linux проектным решениям и, во-вторых, большому числу инструментов, предназначенных специально для этих целей. Часть этих инструментов создана сообществом разработчиков свободного ПО, часть фирмами-производителями ПО.

Эти инструменты и свойства операционной системы мы рассмотрим ниже в подразделах:

- Модульная структура операционной системы
- Механизм обновлений
- Удаленный доступ
- Инструменты для удаленного управления

5.1.3.1 Модульная структура операционной системы

Ядро Linux было спроектировано как модульная структура. Оно не представляет собой монолитный двоичный код, а состоит из небольшой исполняемой части и набора модулей. Модули ядра подгружаются по мере необходимости. Часть модулей загружается во время запуска системы, поскольку без них невозможно обращение к файловой системе и к периферийному оборудованию.

Не только ядро имеет модульную структуру, приложения также построены по модульному принципу. Такие приложения, как языки скриптов (Perl, PHP, Python) или текстовые редакторы (gedit, vi) не интегрированы в операционную системы и в большинстве случаев могут быть заменены другими приложениями или новыми версиями.

Модульный принцип построения дает неоспоримое преимущество при обновлениях и всевозможных заменах, поскольку меняется только то, что необходимо заменить, например, отдельный исполняемый файл, или библиотека, или модуль ядра. Благодаря такому подходу, обновление приложения и даже применение патча, как правило, не требует перезагрузки операционной системы.

5.1.3.2 Механизм обновлений

Как уже упоминалось, обновление приложений не затрагивает работу операционной системы, как правило, его можно запустить параллельно с выполнением других задач. Оно не требует перезагрузки системы, кроме тех случаев, когда обновляется ядро или его модули, и может быть выполнено оперативно.

Что это дает? Кроме упрощения процедуры и сокращения времени, мы имеем также возможность автоматизировать процесс, просто включив процедуру обновления/наложения патчей в расписание автоматически выполняемых задач.

5.1.3.3 Удаленный доступ

Как и для всех Unix-подобных операционных систем, возможность уделенного доступа посредством TCP/IP является для Linux естественной, базовой функцией. Это действительно мощное средство системного администрирования, поскольку все задачи управления можно выполнить удаленно, безопасность при этом обеспечивается использованием шифрованного протокола обмена, например, ssh.

Рассмотрим два типа задач удаленного администрирования:

- Удаленное выполнение административных программ или контролирующих скриптов
- Доступ на клиентскую машину в тех случаях, когда пользователь нуждается в поддержке

Удаленное выполнение. Удаленное выполнение полезно тогда, когда нужно централизованно запустить на клиентских машинах какое-либо приложение или скрипт, например, обновить антивирус или применить свежий патч. Возможно, надо получить информацию от контролирующего скрипта, это поможет вовремя произвести профилактические работы, скажем, очистить каталог с временными файлами, чтобы предотвратить переполнение файловой системы.

Удаленная поддержка. Практически все потребности клиента в поддержке могут быть удовлетворены путем удаленного доступа на клиентскую машину. Единственная проблема, которая может потребовать присутствия технического персонала у клиентской машины — это неработающая сеть. Все остальные вопросы могут быть разрешены удаленно, путем доступа к конфигурационным файлам и журналам на клиентской машине. Такая возможность существенно повышает производительность персонала, отвечающего за техническую поддержку клиентских машин.

5.1.3.4 Инструменты для удаленного управления

Доступно множество инструментов для удаленного администрирования Linux, часть разработана сообществом разработчиков свободного ПО, часть фирмами-производителями ПО. Наиболее популярными из них являются:

- Webmin

  http://www.sourceforge.net/projects/webadmin;

  http://www.webmin.com

- Big Brother/Big Sister

  http://www.sourceforge.net/projects/big-brother;

  http://www.sourceforge.net/projects/bigsister;

  http://www.bb4.org

- Nagios

  http://sourceforge.net/projects/nagios;

  http://www.nagios.org/

Webmin, как правило, используется для удаленного администрирования отдельной машины. Big Brother и Nagios предназначены, в основном, для мониторинга параметров сети в режиме реального времени и выполнения известных административных задач в тех случаях, когда параметры выходят из установленных пределов.

Существуют также и коммерческие инструменты для выполнения подобных задач. Например, компания IBM поставляет пакеты Tivoli® Configuration Manager и Tivoli Monitoring для автоматизации удаленного контроля и администрирования.

Дистрибутивы Red Hat Enterprise Linux используют для удаленного администрирования систем или групп систем Red Hat Network (RHN). RHN содержит сервис для автоматических обновлений, который подробнее описан в Разд. 7.5.7.1.

Дистрибутивы Novell Linux Desktop и SUSE Linux используют пакет для удаленного администрирования систем пакет ZENworks. После того как Novell приобрела программное обеспечение фирмы Ximian, пакет Ximian Red Carpet (называемый теперь ZENworks) стал стандартным средством администрирования, включаемым в вариант «промышленный» дистрибутива SUSE. До этого в дистрибутивах SUSE для удаленного администрирования использовался пакет Yast online Update (YoU).

5.1.4 Настройка системы под нужды пользователя

Иногда бывает полезно специально настроить клиентскую машину под конкретные нужды конечного пользователя. Если стандартная клиентская установка содержит все возможные приложения, которые содержатся в дистрибутиве, то система становится громоздкой, и ее трудно администрировать. Поэтому разумная настройка клиентских машин является существенной частью стратегии развертывания Linux на настольных машинах.

В этом разделе мы остановимся на следующих особенностях клиентской машины под Linux:

- Гибкость в добавлении или удалении компонент стандартной установки Linux
- Контроль пакетов с целью предотвращения «разбухания» клиента
- Возможность построения дистрибутива, ориентированного на конкретные задачи
- Гибкость настройки рабочего стола и администрирования сессии

5.1.4.1 Гибкость в добавлении или удалении компонент

Все дистрибутивы Linux на этапе установки предоставляют пользователю возможность выбрать те компоненты, которые он хочет установить. Также есть возможность использовать стандартный набор пакетов, в этом случае добавить или удалить пакет можно после установки дистрибутива. Эти действия можно выполнить, не прерывая работы системы. благодаря тому, что Linux имеет модульную структуру.

5.1.4.2 Предотвращение «разбухания» клиента

Гибкость в добавлении или удалении компонент помогает предотвратить «разбухание» клиента. Поскольку настройка конкретного клиента достаточно проста, имеется возможность поддерживать несколько вариантов установки системы для конечного пользователя.

Можно собрать работоспособный дистрибутив Linux, который будет помещаться на один гибкий диск (1.4 МБ). При этом стандартный «промышленный» вариант дистрибутива занимает несколько гигабайт.

5.1.4.3 Создание дистрибутивов, ориентированных на конкретные задачи

Поскольку компоненты установки можно свободно исключать, а также есть возможность добавить к ней любое другое приложение, распространяемое свободно, то в некоторых случаях оказывается полезно собрать свой дистрибутив, нацеленный на решение конкретной задачи. Это может быть работа с аудио/видео файлами или специфическая обработка сигналов. Создание собственного дистрибутива может быть полезно в том случае, если в организации есть несколько рабочих мест со специфическими, нестандартными требованиями.

5.1.4.4 Настройка рабочего стола

Различие в подходах между Microsoft Windows и Linux-ориентированным клиентом можно наглядно продемонстрировать на примере менеджера рабочего стола. В Linux существует несколько соответствующих программ, и вы имеете возможность выбрать тот менеджер, который вам больше подходит. Большинство таких программ (например, KDE или GNOME) также предоставляют пользователю средства настройки внешнего вида рабочего стола и интерфейса пользователя. Пользователь может создать свой собственный профиль экрана: задать набор иконок, цвета, фон экрана, вид и управляющие элементы окон и т.п.

5.1.5 Философия открытого ПО

В некоторых случаях уже сам тот факт, что Linux представляет собой открытое ПО, достаточен для того, чтобы принять решение о миграции. Если вы используете свободное ПО, вы не привязаны к конкретной фирме-разработчику. Другой причиной использования свободного ПО может быть желание адаптировать программы под свои специфические нужды. Это всегда возможно, поскольку для свободного ПО всегда доступен исходный код.

Эти причины являются очень весомыми для некоторых правительственных организаций, особенно в Европе.

5.1.6 Простота в эксплуатации

Поскольку сейчас масса рабочих мест оснащена компьютерами, важным свойством ПО, установленного на клиентской машине, является простота его использования. Если для работы с программным обеспечением на обычном рабочем месте необходима длительная интенсивная подготовка персонала, то можно считать, что применение такого ПО требует неоправданных расходов.

5.2 Когда следует мигрировать, а когда нет

Главная цель этой книги — описать процесс миграции клиента на базе Microsoft Window к Linux-клиенту в предположении, что миграция происходит в Windows-ориентированном окружении. На самом деле, миграция

рабочих мест конечных пользователей должна быть частью общего плана перехода предприятия на свободное ПО. Планируя миграцию клиентских машин, не следует упускать из виду это обстоятельство.

Даже если миграция отдельных машин может быть желательна (по причинам, о которых говорится в Разд. 5.1), только общая миграция в итоге даст существенное преимущество. В результате частичной миграции несколько клиентских машин начнет работать под Linux, но при этом общая ИТ-инфраструктура предприятия может стать сложнее.

В этом разделе мы рассмотрим обстоятельства, которые благоприятствуют переводу клиентов на Linux. Коротко говоря, эти обстоятельства сводятся к отсутствию осложняющих факторов. Некоторые из этих факторов мы обсудим более детально.

5.2.1 Функции на клиентской машине соответствуют модели «Тонкий» или модели «Стройный»

Часто клиентские машины, включенные в систему клиент/сервер, в шутку делят на Толстые (fat), Стройные (slim) и Тонкие (thin). Разделение основано на количестве приложений, обычно запускаемых на клиенте. Например, Тонкий клиент запускает локально только интерфейсную сторону приложения, скажем, броузер, а все остальные приложения запускаются на сервере. Различия между этими моделями более подробно описаны в Разд. 7.4.2.

Одним из осложняющих факторов миграции является большое число Толстых клиентов, то есть клиентов, у которых большое число приложений установлено локально. Чем больше число установленных приложений, тем больше шансов, что некоторые из них будет трудно заменить на аналог под Linux. Может случиться, что некоторые из них вообще невозможно перенести, что приведет к дальнейшему усложнению инфраструктуры, подобные случаи описаны в Разд. 7.7.

Если клиент Тонкий, то это означает, что перевод его на Linux не будет являться усложняющим фактором при интеграции в комплексную инфраструктуру. В идеальном случае чтобы получить доступ ко всем необходимым приложениям пользователю нужен только броузер. Тогда миграция под Linux становится просто тривиальной.

Чем больше клиентов соответствуют модели Тонкий/Стройный, тем проще пройдет миграция. Возможный вариант миграции состоит в том, чтобы на первом этапе перевести на Linux только Тонких клиентов, а Толстых оставить в прежнем положении.

5.2.2 Высокое число мигрируемых приложений

Назовем приложение *мигрируемым*, если оно удовлетворяет хотя бы одному из двух требований: либо существует версия данного приложения, работающая под Linux, либо под Linux существует функциональный аналог. То есть, коротко говоря, если есть прямой путь замены этого приложения.

Те приложения, которые не обладают свойством мигрируемости, могут вызвать сложности в процессе миграции, более подробно эти вопросы разобраны в Разд. 7.7. Чем больше мигрируемых приложений используется, тем легче происходит процесс миграции.

Прежде чем планировать миграцию и пытаться справляться с приложениями, которые нельзя перенести, стоит подумать, нет ли возможности отказаться от использования таких приложений и заменить их аналогами, которые допускают перенос между платформами. Такой подход может быть полезен не только в свете миграции на Linux, но и для любых будущих преобразований информационной инфраструктуры.

5.2.3 Организационная готовность

Даже если все остальные факторы благоприятствуют миграции, но организация в целом не готова к этому, попытка миграции будет неуспешной. Организация готова к миграции, если:

- Конечные пользователи этой организации готовы и способны использовать новое программное обеспечение в своей работе
- В организации есть системные администраторы и штат поддержки, которые с энтузиазмом смотрят на перспективу миграции и имеют преставление о той технологии, на которую планируется мигрировать

- Имеется методика разрешения проблем, котрые могут возникнуть во время и после миграции.

Подробнее эти вопросы обсуждаются далее в Гл. 6.

Следует иметь в виду, что наибольшая часть нагрузки, связанной с миграцией, падает на плечи системных администраторов. Именно от этих людей в конечном итоге зависит, будет ли миграция успешной, или она провалится. Энтузиазм системных администраторов и штата поддержки является важнейшим фактором, который надо учитывать, принимая решение о миграции.

5.3 В чем заключаются особенности Linux?

Как уже говорилось в начале Разд. 5.1, возрос о миграции возникает тогда, когда становится доступным альтернативный вариант. И решение о переходе на новую систему основывается на том, что этот альтернативный вариант по своим качествам превосходит используемую сейчас систему. В этом разделе мы в общих чертах опишем основные структурные особенности и уникальные свойства ОС Linux. Далее, в Гл. 8, эти вопросы будут рассмотрены более подробно.

> ЗАМЕЧАНИЕ
>
> Является ли Linux свободным продуктом, или продуктом с открытым исходным кодом или и тем и другим одновременно? Более обстоятельно мы обсуждаем это в Разд. 8.1.1.

На принятие решения о миграции могут влиять различия в работе функций новой системы и существующей (Windows), или другой способ обращения с уже известными функциями, или совершенно новые, уникальные особенности нового окружения. Некоторые такие особенности мы уже упоминали выше в Разд. 5.1 В этом разделе мы будем обсуждать следующие вопросы:

- Много пользователей — зачем?
- Что такое ядро?
- Взаимодействие приложений или одно большое приложение?
- Пользовательский интерфейс, на что обратить внимание?
- Как настольная система взаимодействует с внешней инфраструктурой?
- Каким образом хранятся данные?
- Есть ли другие отличия?

5.3.1 Много пользователей — зачем?

Linux изначально был спроектирован как многопользовательская система. При этом речь шлее не о том: что вычислительной машиной под Linux могут пользоваться несколько человек по очереди, а о реальной многопользовательской системе, когда несколько человек одновременно запускают свои приложения на одном и том же компьютере. Нужно ли это, если речь идет о настольной, персональной машине? Сейчас уже можно твердо сказать «да». Во-первых, настройки и данные, связанные с приложениями, поддерживаются независимо и тщательно для каждого пользователя, а это оказывается существенно при широко применяемом сейчас совместном доступе к приложениям и данным. Во-вторых, для каждого пользователя сохраняются независимо настройки его рабочего стола, каждый раз, выполнив процедуру регистрации, он получает привычное рабочее окружение.

Технические детали, касающиеся многопользовательской архитектуры, можно найти в Разд. 8.2.3. В Разд. 9.3.4 приводится пример прогрессивной стратегии развертывания новых клиентов, в которой интенсивно задействованы многопользовательские возможности Linux.

5.3.2 Что такое ядро?

Основную часть ОС Linux принято называть ядром. В ядро входит самый нижний уровень функций операционной системы, как то: контроль аппаратных средств, запуск драйверов устройств, управление файловыми системами, создание процессов, управление памятью и другие базовые функции. Ядро Linux во многом походит на ядро UNIX.

Ядро Linux имеет некоторые особенности, которые являются совершенно уникальными:

- ядро построено по модульному принципу
- на одном компьютере может быть установлено сразу несколько ядер
- разработка ядра Linux управляется централизовано

Ядро Linux не представляет собой монолитное образование, некоторые его части могут загружаться в процессе работы, такие части называются модулями. Если некоторые функции не требуются в текущий момент, то отвечающие за них модули не загружены и не занимают память.

На одном компьютере может быть установлено сразу несколько ядер, но только одно из них работает в каждый момент времени. Эта особенность позволяет тестировать новые версии ядра, собирать более подходящую его конфигурацию и в то же время иметь возможность очень легко вернуться назад, к старой версии, которая гарантированно работала. Подробнее об этом см. Разд. 8.2.9.

Разработку ядра Linux до сих пор контролирует легендарный Линус Торвальдс. А это означает, что каждая версия ядра представляет собой один единственный объект. Есть разные версии ядра Linux, но ядро версии 2.6.5 в дистрибутиве Red Hat — это в точности то же самое, что и ядро версии 2.6.5 в дистрибутиве Novell.

Информацию о процедуре загрузки ядра можно найти в Разд. 8.2.1.

5.3.3 Взаимодействие приложений или одно большое приложение?

Операционная система Linux далеко не так сильно интегрирована, как операционная система Windows. Фактически каждый дистрибутив состоит из ядра и сотен, если не тысяч отдельных программных пакетов, ассоциированных с этим ядром. Все программы изначально проектировались и проектируются так, чтобы допускать тесное взаимодействие, иначе системы не смогла бы работать. Но это не означает, что они должны взаимодействовать, интеграции между ними может и не быть.

В некоторых случаях такое отсутствие обязательной интеграции на низком уровне может выглядеть как недостаток, но это качество дает разработчикам ПО более высокого свободу в выбора низкоуровневых приложений и возможность их замены впоследствии.

Поскольку ядро Linux и почти все доступные приложения базируются на открытых стандартах, интеграция между ними выполняется легко, и работают они вместе надежно. Подробнее о вопросах стандартизации см. Разд. 8.1.3.

5.3.4 Пользовательский интерфейс, на что обратить внимание?

Как правило, пользователь общается с вычислительно машиной посредством графического монитора, мыши и клавиатуры. Так устроены почти все клиентские компьютеры, хотя внешний вид рабочего стола и функциональность могут различаться.

Под Linux менеджер графической сессии является просто одним из приложений, он не является частью операционной системы. Это означает что, во-первых, вы можете выбирать менеджера рабочего стола (наиболее частый выбор — это KDE или GNOME) и, во-вторых, можно работать вообще без графического окружения, в алфавитно-цифровом режиме. Более подробно эти вопросы обсуждаются в Разд. 8.2.4.

5.3.5 Как настольная система взаимодействует с внешней инфраструктурой?

Настольный компьютер редко работает сам по себе, он должен взаимодействовать с объемлющей инфраструктурой. Он нуждается в подключении к локальной сети, в доступе к серверам и другим общим ресурсам. Linux поддерживает все возможные сетевые протоколы, необходимые для такого взаимодействия.

Важнейшим моментом при подключении компьютера к внешнему миру является проблема обеспечения безопасности. Ядро Linux имеет встроенный брандмауэр, который защищает компьютер от несанкционированного доступа извне и обеспечивает безопасную работу конечных пользователей и сохранность данных. Различия в подходах различных ОС к вопросам безопасности уже обсуждались выше, в Разд. 5.1.1.

5.3.6 Каким образом хранятся данные?

Способ хранения данных в Linux кардинально отличается от способа, принятого в Windows. Файловая структура в Linux представлена одним деревом, при этом различные типы разделов, в том числе разделы на удаленных устройствах, выглядят однотипно. Здесь нет букв, приписанных к дискам. Такой подход позволяет, например, придерживаться одной и той же логической структуры каталогов на всех клиентских машинах. Подробнее об этом см. Разд. 8.2.6.

Существенным отличием в обращении с файлами является существование ссылок в большинстве файловых систем, с которыми работает Linux. По сути ссылки являются указателями на файлы или на целые каталоги. Существуют два вида ссылок: жесткие ссылки и символьные ссылки. Жесткая ссылка является просто еще одним именем, связанным с файлом, а символьная ссылка — это отдельно стоящий указатель. В том случае, если удален файл, символьная ссылка на него не удаляется, но начинает указывать в пустоту. Если число жестких ссылок на файл больше одной, то удаление одного имени не повлечет удаление файла. Реально он будет удален только после удаления последней жесткой ссылки. Подробнее об этом можно прочесть в Разд. 8.2.8.

5.3.7 Есть ли другие отличия?

Есть и другие особенности, отличающие Linux от других ОС для настольных компьютеров. В этом разделе мы рассмотрим разницу в работе с виртуальной памятью и уникальное для Linux понятие уровней выполнения (run levels).

Работы с виртуальной памятью в каждой операционной системе происходит по своему, иногда это зависит даже от версии операционной системы. Особенностью Linux является то, что виртуальная память не будет использоваться до тех пор, пока есть возможность работать в реальной оперативной памяти. Windows, например, начинает перемещать информацию из оперативной памяти на диск и в других случаях, там существует практика превентивного свопинга. Во многих случаях такой подход приводит в снижению скорости выполнения операций. В Linux принята также система кэширования, то есть, хранение недавно использованной файловой информации в оперативной памяти. В результате такой практики постоянно используется значительная часть оперативной памяти. В том случае, если дополнительная оперативная память требуется приложениям, система просто сокращает область кэширования. Подробнее об использовании виртуальной памяти см. Разд. 8.2.7.

Концепция уровней выполнения является общей для UNIX/Linux-подобных операционных систем. Уровень выполнения определяет, какие системные сервисы будут запущены при начальной загрузке системы. Уровни выполнения нумеруются от 0 до 9. Например, уровень выполнения 3 соответствует загрузке всех системных сервисов, кроме графических. На уровне 5 стартует также и графическое окружение. Уровень 1 соответствует однопользовательскому режиму загрузки системы, в котором доступна олько одна консоль и отключены практически все службы. Подробнее об уровнях выполнения см. Разд. 8.2.5.

5.4 Цели миграции

Каждый случай миграции может иметь свои особенности. Конечная цель миграции, естественно, должна состоять в том, чтобы перевести все настольные машины на Linux. Однако, для некоторых организаций разумным подходом будет начать с пилотного проекта и на начальном этапе осуществить миграцию только части клиентских машин.

В этом разделе мы обсудим более детально оба подхода, обсудим различия в целях и задачах полной миграции и пилотной миграции. Что же касается технических деталей планирования и осуществления частичной, пилотной или полной миграции, то им и посвящена, собственно, вся оставшаяся часть книги.

5.4.1 Пилотная миграция

Формально целью пилотной миграции является перевод нескольких настольных машин на работу под Linux, однако фактически главная цель пилотной миграции — проверить правильность расчетов перед полной миграцией и завоевать доверие персонала.

Пилотную миграцию надо планировать так, чтобы на новых рабочих местах можно было испытать все виды деятельности персонала, которые будут включены в полную миграцию. В пилотной миграции должны быть задействованы все приложения, которые будут использоваться впоследствии. В этом смысле пилотную миграцию можно рассматривать как испытание, как средство сбора данных для коррекции планов перед полной миграцией.

Пилотная миграция должна ответить на следующие вопросы:

- Все ли приложения адекватно работают после миграции?
- Может ли персонал после миграции так же эффективно использовать информационную инфраструктуру предприятия, как и до миграции?
- Не усложнились ли после миграции задачи администрирования?
- Все ли компоненты инфраструктуры, такие как терминальные серверы и консолидированные настольные машины под Windows, функционируют так, как планировалось?

5.4.2 Полная миграция

Используя те методы, которые были опробованы и проверены во время пилотной миграции, можно уверенно совершить полную миграцию, то есть перевод всех намеченных рабочих станций на новую операционную систему.

Полная миграция может иметь следующие цели:

- Повысить защищенность рабочих станций
- Улучшить управляемость клиентских машин
- Понизить полную стоимость содержания соответствующих рабочих мест
- Снизить зависимость от отдельного производителя ПО
- Продлить жизненный цикл компьютерного оборудования
- Соответствовать государственным правилам или стратегии (например, Китай декларировал использование свободного ПО как стратегический императив)
- Распространить полезный опыт использования Linux на серверах на рабочие станции
- Как предлог для наведения порядка в своем хозяйстве (например, стандартизовать рабочие места и таким образом освободиться от груза накопившихся проблем)

Хорошее предварительное планирование — важнейшая часть миграции. Если вы начинаете работу с ясными перспективами, вам проще будет справиться с техническими трудностями. Поддержка персонала также очень важна для успешной миграции.

Ниже в этой книге мы подробно обсудим и то, как планировать техническую сторону миграции, и то, как правильно учитывать человеческий фактор. Затем будет приведен пример миграции, включающий подробный разбор путей решения некоторых технических проблем.

Часть III

Планирование пилотной миграции

Глава 6

Организационное планирование и человеческий фактор

Процесс замены программного обеспечения для конечных пользователей имеет существенные отличия от обновления ПО серверной части. Во втором случае вы имеете дело, главным образом, с техническими проблемами, а при замене клиентской части на первый план выходят организационные моменты и трудности, связанные с человеческим фактором, поскольку такие изменения затрагивают повседневную работу большого числа людей. Именно эти вопросы и будут обсуждаться ниже.

Данная глава содержит следующие разделы:

- Разд. 6.1 Изучение паттернов использования компьютеров

  Изучение, выделение и оценка паттернов использования компьютеров на рабочих местах, где под паттерном мы понимаем модель или образец рабочего места, совокупность установленных форм, правил и обычных, привычных для персонала действий

- Разд. 6.2 Обеспечение функциональной преемственности

  О том, что понимается под термином «функциональная преемственность», и какие существуют способы ее обеспечения в процессе миграции

- Разд. 6.3 Человеческий фактор

  Как правильно спланировать работу с персоналом в переходный период

- Разд. 6.4 Вопросы переобучения

  Как минимизировать расходы и сократить время на переобучение конечных пользователей

6.1 Изучение паттернов использования компьютеров

Одной из важнейших задач планирования пилотной миграции является анализ текущей информационной среды, нацеленный на выделение паттернов использования компьютеров конечными пользователями. Благодаря такому анализу у вас появится список паттернов, которые будет необходимо реализовать. Так вы сможете эффективно разделить общую задачу на простые составляющие. Хорошим способом сбора информации о паттернах является опрос конечных пользователей. Пример методики подобного опроса приведен в Разд. 7.1.

6.1.1 Разделение рабочих мест на функциональные группы

Для удобства дальнейшего администрирования полезно разделить клиентские машины по функциональным группам в соответствии с теми обязанностями, которые возложены на сотрудника на данном рабочем месте, и задачами, которые ему приходится решать. Ниже показан возможный тип функционального разбиения рабочих мест на пять групп: от фиксированных функций до продвинутого офиса.

Опишем каждую из групп подробнее:

<table>
<tr><td>Ограниченная функциональность</td><td>Техническая рабочая станция</td><td>Рабочая станция делопроизводителя</td><td>Базовый офис</td><td>Стандартный офис</td></tr>
<tr><td colspan="2">Ограниченная использование бизнес-приложений</td><td colspan="3">Полный набор приложений для управления бизнес-процессом</td></tr>
<tr><td>Упрощенный офис</td><td colspan="3">Простой офис</td><td>Продвинутый офис</td></tr>
<tr><td>Отсутствует электронная почта</td><td colspan="3">Постой вариант электронной почты</td><td>Продвинутый вариант электронной почты</td></tr>
<tr><td>Нет непосредственного обмена сообщениями</td><td colspan="4">Непосредственный обмен сообщениями</td></tr>
<tr><td colspan="4">Простой доступ к интранету посредством броузера, порталы</td><td>Продвинутый доступ к интернету посредством броузера</td></tr>
<tr><td colspan="5">Система управления файлами и печатью, доступ к локальной сети, эмуляция хостов</td></tr>
</table>

Рис. 6.1: Пример разбиения рабочих мест в соответствии с ролью работника

Рабочее место с фиксированными функциями

Пользователи такой рабочей станции запускают только фиксированный и ограниченный набор приложений. Приложения могут быть настроены на выполнение специфических функций. Примерами могут служить кассовый терминал или киоск-терминал.

Техническая рабочая станция

Рабочее место со специфическими производственными функциями. Как правило, на таких рабочих станциях установлен ряд специфических, инженерных пакетов. Примерами может служить рабочее место программиста, проектировщика, рабочее место для промышленной обработки изображений.

Рабочая станция делопроизводителя

Рабочее место сотрудников, существенным моментом в работе которых является обмен данными. Здесь необходимы доступ к локальной сети, доступ к интернету, возможность обмена электронными сообщениями. Примерами могут служить рабочее место сотрудника справочной службы, рабочее место по приему заказов, диспетчера или сотрудника отдела логистики.

Базовый офис

Рабочее место сотрудника, занимающегося организацией бизнес-процесса. Здесь могут быть установлены специфические приложения, предназначенные для планирования и управления ресурсами предприятия, управления взаимоотношениями с клиентами и т.п. Требуется доступ к локальной сети, доступ к интернету, возможность обмена электронными сообщениями, в том числе непосредственный обмен сообщениями и электронная почта с вложениями. Требуется простой офисный пакет, позволяющий создавать и просматривать текстовые документы и электронные таблицы в основном внутреннего пользования. Требуется возможность создавать документы в переносимых форматах, например, PDF, RTF или HTML. Примерами таких рабочих мест могут быть рабочее место банковского служащего или администратора офиса небольшого предприятия.

Стандартный офис

Рабочее место, на котором решается широкий круг задач: от приложений, обеспечивающих бизнес-процесс, аналогичных приложениям для базового офиса, до создания сложных документов типа презентаций или оригинал-макетов публикаций. Для такого рабочего места характерно интенсивное использование локальной сети и файлов общего доступа. Здесь должны быть установлены средства личного делового планирования, обмена информацией вплоть до видеоконференций. Доступ в интернет должен допускать широкое использование мультимедийных средств. Примерами таких рабочих мест могут быть рабочее место руководителя проекта, маркетолога, финансового менеджера, руководителя крупного и среднего бизнеса.

ПОДСКАЗКА

Чем левее в изображенной на рисунке таблице находится рабочее место, предназначенное для миграции, тем дешевле и проще будет происходить на нем процесс перехода. Прежде чем планировать миграцию рабочих мест, относящихся к самому правом сегменту, надо хорошо взвесить все за и против. Следут принимать во внимание, что набор и функциональность приложений для Linux постоянно расширяются. Возможно лучшим решением будет отложить миграцию наиболее продвинутых клиентов на некоторое время.

6.1.2 Описание профилей рабочих мест

После того как выполнено разделение рабочих мест на функциональные группы, следует создать описание профиля каждой группы. Профиль рабочего места включает список приложений, используемых на данном рабочем месте, описание зависимости между приложениями (например, наличие обмена данными, связь с сервером базы данных), список используемых типов файлов, описание аппаратных средств и периферийного оборудования, включая взаимозависимость периферийных устройств.

При обследовании аппаратных средств и периферийного оборудования следует обратить особое внимание на вопросы совместимости. Также крайне полезно составить список всех типов используемых файлов и указать, какие приложения работают с каждым типом файлов. По крайней мере, в список надо включить все широко распространенные в компании типы. Работа со списком типов файлов служит гарантией удобства и простоты использования данных после миграции.

6.1.3 Опрос конечных пользователей

Опрос конечных пользователей может потребовать много времени, но, тем не менее, разумно будет его провести, поскольку это позволит посмотреть на ситуацию изнутри, глазами персонала. Вы сможете уточнить, каким образом конечные пользователи взаимодействуют с информационной системой, и что наиболее важно для них.

Есть один важный момент, который можно выяснить только в процессе опроса персонала, а именно — наличие на клиентских машинах разнообразных средств, не включенных в каталог используемого в компании ПО. Часто пользователи устанавливают собственные надстройки, шаблоны, держат свои базы данных, а иногда и устанавливают дополнительные приложения, которые помогают им в повседневной работе.

6.2 Обеспечение функциональной преемственности

В этой главе описываются методы, которые следут применять в процессе миграции для поддержания функциональности уже работающих приложений. Что ждет пользователей после завершения процесса миграции? В большинстве случаев они переключатся на работу с новыми, но функционально эквивалентными приложениями. Иногда удается использовать прежние программы на новой платформе, некоторые приложения придется перенести на терминальный север. Следует заранее продумать стратегию перевода пользователей на новую платформу, это поможет избежать снижения производительности в переходный период.

6.2.1 Приложения, работающие на обеих платформах

Для некоторых приложений, родной средой исполнения для которых является Linux, существуют версии, работающие под Windows. Использование таких приложений позволяет минимизировать проблемы переходного периода и обойтись без переучивания персонала. Вполне можно постепенно заменить ряд приложений на их функциональные аналоги, работающие также и под Linux, до того, как будет заменена операционная система.

Поскольку пользователи работают в знакомой среде, а новые приложения функционально будут похожи на старые, такое изменение не вызовет у них никаких трудностей.

Затем, после проведения миграции операционной системы, ситуация повторится, только в этом случае среда станет новой, а приложение останется старым, уже знакомым, поэтому освоиться с нововведениями будет проще.

Табл. 6.1 содержит примеры приложений, работающих под Microsoft Windows, вместе с их функциональными аналогами, работающими под управлением обеих операционных систем. Использование этих аналогов позволяет начать миграцию приложений до миграции операционной системы.

Таблица 6.1: Примеры приложений, работающих на обеих платформах

Приложение под Windows	Linux-ориентированный функциональный аналог
Microsoft Outlook, Outlook Express	Mozilla, Mozilla Thunderbird, Evolution
Microsoft Word	OpenOffice.org Writer
Microsoft Excel®	OpenOffice.org Calc
Microsoft Powerpoint	OpenOffice.org Impress
Jasc Paint Shop Pro, Adobe Photoshop	The GIMP
Messenger-Client (MSN®, Yahoo, ICQ, AIM)	GAIM
IBM Lotus Sametime® 7.5	Исходно программа для Windows, поддерживается Linux
IBM Lotus Notes 7.x	Исходно программа для Windows, поддерживается Linux

Сетевые ресурсы, связанные с проектами:

- OpenOffice.org:

  http://www.openoffice.org

- The GIMP:

  http://www.gimp.org

- GAIM:

  http://gaim.sourceforge.net

- IBM Lotus Sametime® 7.5:

  http://www-142.ibm.com/software/sw-lotus/products/product3.nsf/wdocs/st75home

- IBM Lotus Notes 7.x

  http://www.ibm.com/software/sw-lotus/products/product4.nsf/wdocs/linux

ЗАМЕЧАНИЕ

IBM Lotus Sametime и IBM Lotus Notes будут поддерживать родные версии и для Windows, и для Linux, начиная с версий IBM Lotus Sametime 7.5 и IBM Lotus Notes 7.x.

6.2.2 Функциональные эквиваленты служебных приложений

Приложения обслуживающего характера, такие как средства работы с файловой системой, архиваторы и средства просмотра файлов, более тесно связаны с управляющей операционной системой. Как правило, они не имеют версий, работающих на разных платформах, поэтому их можно начать использовать только после миграции ОС, то есть с ними нельзя поступить так же, как с приложениями, описанными в предыдущем разделе.

Ниже приводится список функциональных аналогов служебных программ. Обратите внимание, что в каждом случае есть выбор между несколькими приложениями, и при этом любой вариант предоставляет не меньший набор функций, чем приложение Windows, а зачастую и больший.

Таблица 6.2: Функциональные аналоги служебных программ

Файловые менеджеры (такие как Windows Explorer)	Konqueror (KDE), Nautilus (GNOME)
Архиваторы (такие как WinZip)	Karchiver (KDE), Ark (KDE), FileRoller (GNOME)
Средства просмотра файлов (такие как Adobe Reader)	Konqueror (KDE), KView (KDE), Nautilus (GNOME), Evince (GNOME), Adobe Reader for Linux
Мультимедиа проигрыватели (такие как Windows Media® Player)	xmms, xine, RealPlayer, totem, amaroK (KDE), JuK (KDE), Rhythmbox (GNOME), Banshee (GNOME)

6.2.3 Веб-приложения

К сожалению, не для всех необходимых приложений можно найти близкие по функциям или переносимые варианты. Приложения для ERP (планирование и управление ресурсами предприятия), CRM (управление взаимоотношениями с клиентами) или средства для организации совместной работы, как правило, имеют Толстую клиентскую часть и не имеют кросс-платформенных вариантов, работающих и под Windows, и под Linux. Как мы можем видеть, производители промышленного ПО не пытаются решить эту проблему путем создания отдельных клиентов для каждой операционной системы, они переходят на использование клиентов, работающих посредством веб-броузеров, ориентируясь на платформо-независимые броузеры.

Такой подход, безусловно, является шагом вперед, поскольку приложение с веб-ориентированным интерфейсом никогда не будет зависимо от конкретной операционной системы.

Хорошим примером подобного подхода является пакет Domino Web Access, ранее известный как iNotes. До версии 6.5 iNotes работал только с Internet Explorer и поэтому был доступен только под управлением Windows. Более поздние версии работают с броузером Mozilla, благодаря чему приложение стало кросс-платформенным. Продукты компании SAP являются еще одним примером подобного подхода. Они обеспечивают полнофункциональную клиентскую часть на основе веб-интерфейса и написанных на Java Толстых клиентов, работающих с платформо-независимыми броузерами.

Если же веб-ориентированного решения не существует, простая миграция такого приложения в новое рабочее окружение через кросс-платформенный клиентский веб-интерфейс становится невозможной. В этом случае следует попытаться найти более свежие версии используемого программного обеспечения, допускающие работу посредством кросс-платформенных клиентов. Если таких версий нет, то, скорее всего, придется переключиться на близкое по функциям ПО другого производителя, соответствующее требованиям кросс-платформенности.

6.2.4 Сервер-ориентированная обработка данных

Помимо рассмотренных в предыдущих трех разделах, существует еще один подход, который позволяет облегчить конечным пользователям переключение на новое окружение. Этот подход состоит в том, чтобы снизить «массу» клиентов путем перенесения функциональности приложения с клиента на сервер, что подразумевает использование терминальных серверов.

Содержание функциональной части на сервере само по себе дает ряд преимуществ, например, упрощение процедуры обновления или более рациональное использование мощностей компьютеров. Но главное преимущество такого подхода состоит в том, что интерфейсная часть отделена от функциональной. Независимо от того, какая операционная система установлена на терминальных компьютерах и какой используется тип клиента, предоставляемые сервером приложения имеют один и тот же интерфейс. Единственным предъявляемым к терминалу требованием является наличие приложения, обеспечивающего установку стабильного и безопасного соединения с терминальным сервером.

Можно использовать Windows Terminal Server или Citrix Metaframe, выбирая для них те клиенты, которые работают одновременно и под Windows, и под Linux. Для доступа к собственному терминальному серверу MS Windows из-под Linux можно использовать rdesktop, клиент с открытым кодом, использующий протокол RDP (remote desktop protocol). Найти этот весьма полезный инструмент вы можете по адресу:

http://sourceforge.net/projects/rdesktop/

Citrix поставляет вместе с пакетом MetaFrame Suite ICA клиенты для ряда операционных систем, включая ICA Client для Linux.

Использование такого подхода позволяет переместить многие приложения на серверную сторону, соответственно, процесс миграции будет включать процедуру перехода от Толстых клиентов к Тонким. Наиболее продуктивным подобный переход будет для рабочих мест, соответствующих паттернам «Фиксированные функции» и «Рабочая станция делопроизводителя», Рис. 6.1 содержит их описание.

Новый подход к сервер-ориентированной обработке данных использован фирмой NoMachine при создании сервера NX. Использование этого сервера значительно снижает сетевой трафик как при работе с X-Window, так и при использовании RDP. Сервер NX выпущен под лицензией GPL, и в ближайшее время ожидается появление реализаций клиентов для различных платформ. Дополнительную информацию по NX можно найти на веб-странице по адресу:

http://en.wikipedia.org/wiki/NX_technology

http://www.nomachine.com

6.3 Человеческий фактор

Планируя миграцию клиентского ПО, крайне важно продумать стратегию работы с персоналом, поскольку изменения в данном случае будут происходить главным образом на рабочих местах конечных пользователей.

Естественно ожидать, что изменение привычного интерфейса на рабочем компьютере вызовет различную реакцию персонала — от энтузиазма по поводу нововведений до полного неприятия. Здесь важно в первую очередь держать конечных пользователей в курсе предстоящих перемен.

Рис. 6.2 демонстрирует один из вариантов изменения отношения персонала к миграции.

Рис. 6.2: Кривая изменения отношения персонала к миграции

Хорошо составленный план взаимодействия позволит сгладить кривую и ускорить переход к положительному восприятию перемен. Разумный обмен информацией вместе с соответствующим переобучением позволят минимизировать количество пользователей, которым не понравятся нововведения и увеличить число пользователей, которые примут перемены без недовольства уже на начальном этапе и ускорят переход к активному усвоению и производительному использованию.

Эти проблемы могут быть наиболее актуальны при работе с персоналом службы технической поддержки и сопровождения. Дело в том, что персонал ИТ-отдела может воспринять решение руководства о переключении на другую операционную систему как неудовлетворительную оценку своей текущей деятельности. Кроме того, могут сыграть отрицательную роль опасения, связанные с отсутствием опыта работы в новой системе. Согласитесь, будет непросто убедить системного администратора Windows в том, что переход на Linux-клиенты — это хорошая стратегия. Необходимо прежде убедить его в том, что в таком переходе нет ничего личного, и что организация готова затратить на его переподготовку достаточные средства.

Таким образом, прежде чем начинать миграцию, необходимо сосредоточить внимание на двух важных аспектах:

- Разработка плана взаимодействия
- Выделение группы, наиболее подходящей для пилотной миграции

6.4 Вопросы переобучения

В процессе миграции во многих случаях нельзя обойтись без переобучения персонала. Любые курсы переобучения требуют значительных расходов, во-первых, сами по себе, во-вторых, за счет отрыва персонала от основной работы. В этом разделе мы рассмотрим вопросы переобучения и пути сокращения расходов на переобучение.

6.4.1 Кросс-платформенные приложения помогут отделить переобучение от миграции

Как уже обсуждалось в Разд. 6.2.1, применение стратегии постепенной замены пользовательских приложений на кросс-платформенные приложения позволяет персоналу постепенно освоиться с нововведениями. При этом, естественно, сокращаются и расходы на переобучение. Для работы с новым приложением в знакомой среде пользователю достаточно небольшого инструктажа. С другой стороны, после замены клиентской операционной системы у сотрудников не возникнет трудностей, поскольку работа будет вестись с уже знакомыми приложениями.

6.4.2 Сохраняем визуальный интерфейс

Еще один подход, позволяющий облегчить переход на новую систему и, соответственно, минимизировать затраты на переобучение персонала, состоит в том, чтобы максимально сохранить визуальный интерфейс пользовательских приложений и рабочего стола. Вполне можно так настроить Gnome или KDE, что различия в визуальном интерфейсе рабочего стола и в способе запуска приложений по сравнению с рабочим столом Windows Desktop будут минимальными. Существует множество соответствующих свободно распространяемых профилей, которые можно загрузить и донастроить для своих нужд.

Примеры таких профилей можно найти на страницах:

http://www.kde-look.org

http://art.gnome.org

6.4.3 Привычные действия

При настройке интерфейса полезно обратить внимание на то, что в разных системах приемы выполнения некоторых простых действий могут отличаться. Например, в Windows окна открываются двойным щелчком мыши, а в Linux — одинарным. Хорошей идеей будет перенастроить эту функцию для пользователей, привыкших к Windows.

6.4.4 Файловая система: сохранить структуру подкаталогов

Пользователи Windows привыкли к тому, что на верхнем уровне иерархической файловой системы находятся точки монтирования разделов, такие как C:\, D:\ и т.д. В Linux применяются несколько иные соглашения об организации файловой системы. Общепринятые точки монтирования в файловых системах под Linux включают `/usr`, `/home` и т.п. После миграции у конечных пользователей могут возникнуть трудности с поиском файлов в новой файловой системе. Чтобы облегчить пользователям процесс перехода к новой иерархии, имеет смысл перенести все содержимое пользовательского каталога `My Documents` в каталог с таким же названием в домашнем каталоге пользователя `/home/user` в системе Linux. В каталоге `/home/user/My Documents` следует сохранить существовавшую в Windows структуру подкаталогов и их содержимое.

6.4.5 До миграции: просто попробуем поработать с Linux

В мире Linux существует такая полезная вещь, как «живой» дистрибутив. Живым дистрибутивом называют загружаемую с компакт-диска операционную систему, которая может работать без установки на жесткий диск. Одним из наиболее популярных живых дистрибутивов является Knoppix, информацию о котором можно найти по адресу:

http://www.knoppix.com

Что такое живой дистрибутив? Приведем описание со страницы Knoppix:

> KNOPPIX представляет собой загрузочный компакт-диск, содержащий набор программного обеспечения GNU/Linux, инструменты для автоматического распознавания аппаратных средств, поддержку большого числа видеокарт и звуковых карт, SCSI и USB устройств и другой периферии. KNOPPIX можно использовать для демонстрации возможностей Linux, как средство обучения, в качестве мощной «аварийной дискеты» или как платформу для запуска демо-версий коммерческого программного обеспечения. Нет необходимости устанавливать что-либо на жесткий диск, примерно 2 ГБ данных находятся на CD в сжатом виде, при этом разархивирование происходит оперативно во время исполнения.

Работа с живым дистрибутивом даст возможность пользователям опробовать и оценить особенности интерфейса, возможности приложений и другие аспекты Linux-клиента, прежде чем произойдет реальная миграция. Все это никак не затронет работоспособность операционной системы, установленной на жестком диске компьютера. Использование живого дистрибутива при подготовке к миграции полезно еще и тем, что позволяет превентивно выявить возможные проблемы с аппаратным обеспечением и периферией. Живой дистрибутив поможет заранее подобрать все драйверы, необходимые для поддержки аппаратной конфигурации.

Глава 7

Техническое планирование

В начальных главах мы рассматривали в основном стратегические и организационные вопросы миграции, во многом абстрактные, это был взгляд сверху.

Теперь пора спуститься с неба на землю и обсудить технические детали процесса перехода от Windows-клиента к Linux-клиенту.

Эта глава содержит следующие разделы:

- Разд. 7.1 Изучение имеющейся информационной среды
- Разд. 7.2 Взаимодействие с имеющимися сетевыми службами
- Разд. 7.3 Стандартизация рабочего стола
- Разд. 7.4 Планирование миграции приложений
- Разд. 7.5 Планирование управления Linux-клиентом
- Разд. 7.6 Сравнение процедуры миграции настольных машин и ноутбуков
- Разд. 7.7 Приложения, не допускающие переноса под Linux
- Разд. 7.8 Развертывание новых клиентов
- Разд. 7.9 Выявление неисправностей и техническая поддержка после миграции

7.1 Изучение имеющейся информационной среды

Прежде чем начинать планировать технические детали будущего окружения, в котором будет работать Linux-клиент, крайне важно изучить текущее состояние информационной среды и процессы, связанные с ее сопровождением.

ИТ-инфраструктура включает, помимо рабочих станций, множество других компонент, таких как серверы печати и файл-серверы, серверы баз данных, службы поддержки каталогов, серверы приложений (как локальные, так и удаленные) вплоть до мейнфреймов, к которым рабочие станции получают доступ через веб-интерфейс или посредством API. Некоторые вопросы, связанные с функционированием сети, более подробно будут рассмотрены в Разд. 7.2.

Поскольку мы рассматриваем миграцию рабочих станций конечных пользователей, для нас наибольшее значение будут иметь две области ИТ-инфраструктуры:

- Взаимодействие между людьми: изучение паттернов использования компьютеров
- Техническая сторона: свойства самих настольных систем, специальные устройства, внешние требования, связанные с интеграцией

7.1.1 Оценка общей ситуации

Прежде чем углубляться в какие-либо технические детали, необходимо уточнить, какие именно требования сформулированы для данного проекта миграции. Правильное понимание целей миграции в рамках общей стратегии развития информационных технологий предприятия является важным фактором успешного выполнения миграции.

Вопросы, которые полезно задать для уточнения общей ситуации:

- Каковы тактические задачи миграции?
- Каковы критерии успеха при решении каждой из этих задач?
- Каковы финансовые аспекты миграции (бюджет, ожидаемый доход от вложений, оценка потребностей и т.п.)
- Каковы стратегические цели миграции? Как миграция вписывается в общую стратегию развития ИТ-структуры предприятия?

7.1.2 Инвентаризация оборудования

Следует составить описание оборудования на всех рабочих местах, на которых планируется миграция. Такое описание необходимо для того, чтобы предвидеть все возможные проблемы с поддержкой оборудования, чтобы определить, что необходимо заменить. Кроме того, подобное описание будет полезно при планировании обновлений оборудования в будущем.

В идеальном варианте подобное описание должно делаться автоматически, регулярно. Это позволит иметь историю обновлений оборудования и осуществить единовременный прогон при составлении описания перед миграцией. Технология eGather и Asset Depot, предлагаемые фирмой IBM, могут помочь в этом деле. Их подробное описание содержится в документе IBM Redpaper *Using Asset Depot for Inventory Management*, который можно найти интернете по адресу:

http://www.redbooks.ibm.com/abstracts/redp3763.html?Open

Вот список вопросов, ответы на которые надо получить в процессе инвентаризации оборудования:

- Какое оборудование используется (производитель, тип, модель)? Если в результате возникает большой, неупорядоченный список, то стоит подумать об объединении части пунктов по какому-либо признаку, который может зависеть от используемой платформы.
- Стандартизовано ли оборудование? Если все машины имеют один и тот же тип, то обеспечение поддержки периферии и переустановка дистрибутивов значительно упрощаются, как в текущей ситуации, так и при переходе на Linux.
- Какие внешние устройства локально подключены к каждому компьютеру, учитывая только те, которые реально используются? Этот список должен включать все принтеры, сканеры, специальные типы клавиатур, необходимые для делового обмена данными и т.п. Он не должен включать оборудование, установленное просто для забавы или потому, что пользователь является важной персоной, и на его столе из-за этого должно стоять что-нибудь особенное.
- Было ли включено требование поддержки Linux в заявку при покупке оборудования?
- Все ли аппаратные средства, установленные на рабочем месте, реально используются в работе? Например, на компьютере может быть установлена звуковая карта, но при этом не установлен драйвер для управления ею, так как на данном рабочем месте не предусмотрено использование звука. В таком случае при установке другой операционной системы поддержка звуковой карты также не потребуется.

IBM предоставляет информацию о текущих моделях производимого оборудования, таким образом каждый, кто использует оборудование фирмы IBM, может проверить, поддерживается ли оно официально, и найти необходимые драйверы. Информацию можно найти на странице:

http://www-306.ibm.com/pc/support/site.wss/MIGR-48NT8D.html

В Табл. 7.1 приведен пример списка, получившегося в результате инвентаризации клиентского оборудования. В графе «Периферия» приводится список всех подключенных периферийных устройств, как то: принтеры, сканеры, цифровые фотокамеры, USB-устройства и т.д.

Таблица 7.1: Пример инвентаризационного списка клиентского оборудования

Модель	Тип	RAM в ГБ	Диск в ГБ	Всего	Периферия
2373MU4	T40	2	80	120	
23668AU	T30	1	60	42	
2366MU1	T30	1	48	1023	
2366MU9	T30	0.5	48	311	
26473U5	T23	1	48	278	
26476RU	T23	0.5	30	67	
26478EU	T22	1	30	44	
264757U	T21	0.5	24	1	
2682MU3	R40	1	48	99	

7.1.3 Инвентаризация программного обеспечения

Следует составить описание программного обеспечения, установленного на всех рабочих местах, на которых планируется миграция. При разработке плана миграции такое описание поможет составить список приложений и сервисов, учесть специфические конфигурации и другие особенности. Составляя подобное описание, надо обратить внимание и на те вопросы, которые не лежат на поверхности, например, на задачи, которые запускаются автоматически по расписанию, или пользовательские ID, заведенные со служебными целями.

Вот список ключевых вопросов:

- Каков набор приложений, поставленных внешними продавцами или производителями ПО? В результате ответа на этот вопрос должен возникнуть список приложений, включающий производителя, текущую версию продукта и приложенные к нему патчи.
- Каков набор приложений, разработанных внутри компании? В результате ответа на этот вопрос должен возникнуть список приложений, разработанных и поддерживаемых непосредственно техническими службами компании.
- Каков набор приложений, требующих доступа к внешним данным? В результате ответа на этот вопрос должен возникнуть список приложений, которые обращаются к файловым серверам, серверам приложений, веб-серверам, базам данных, мэйнфреймам и другим внешним ресурсам. Подробнее см. Разд. 7.4.
- Определены ли группы пользователей? Чем характеризуется каждая группа? По какому признаку сгруппированы пользователи? Группирование может быть сделано по месту работы, то есть отделу, где работает сотрудник, используемым приложениям, типу выполняемых обязанностей или уровню ответственности.
- Каков набор приложений, отвечающих за безопасность, и каковы правила регуляции доступа на данном рабочем месте? В этот список должны попасть антивирусные программы, средства идентификации пользователя и блокировки экрана, должно быть указано, включен ли брандмауэр, и открыт ли доступ к портам, а также перечислены правила установки, сопровождения и обновления приложений, применения патчей к компонентам операционной системы и установленным приложениям.

7.1.4 Описание форматов данных и их взаимозависимостей

Для большинства приложений типа клиент/сервер не возникает никаких проблем с форматами данных, для них единственным условием плавной миграции является наличие клиента, который может работать под управлением Linux. Это справедливо для большинства баз данных, а также и для приложений, которые используют веб-интерфейс для доступа к данным на сервере.

Тем не менее, существует ряд приложений, как правило локальных и специализированных, которые используют собственные форматы данных, в худшем случае эти форматы могут иметь зависимость от файловой системы. Подобные случаи могут потребовать специальных процедур для конвертирования данных в приемлемый формат.

В эту категорию приложений попадают некоторые почтовые клиенты (например, Lotus Notes) или офисные пакеты (Lotus Smartsuite или Microsoft Office). В процессе сбора информации о текущем положении дел не стоит обсуждать технические детали миграции, следует просто учесть, какие из подобных приложений используются.

Вот приблизительный список вопросов, которые следует задать:

- Используется ли Microsoft Office? Если да, то какие компоненты и как часто?
- Используются ли макросы в Microsoft Office? Если да, то какие типы макросов и как часто?
- Используется ли Microsoft Outlook? Если да, то какие компоненты и как часто?
- Используется ли Lotus Smartsuite? Если да, то какие компоненты и как часто?
- Используется ли Lotus Notes? Если да, то какие компоненты и как часто?
- Используется ли Microsoft Project? Если да, то какие компоненты и как часто?
- Используется ли Visual Basic для автоматизации действий внутри отдельных приложений или при обмене данными между приложениями?
- Есть ли обмен файлами с внешними организациями? Если да, то как часто и каковы форматы файлов обмена?

Получив ответы на эти вопросы, вы не только узнаете, какие клиенты используются конечными пользователями, но и поймете, какие продукты лежат в основании информационной инфраструктуры. Например, если в качестве клиента используется Microsoft Outlook, то роль сервера, скорее всего, играет Microsoft Exchange, если клиентом является Lotus Notes, то на сервере, видимо, установлено Lotus Domino. В обоих случаях при планировании миграции клиентских машин нужно принимать во внимание то обстоятельство, что на серверной стороне установлены подобные продукты.

Ниже приведена таблица, в которой показано, какие приложения могут быть заменены функциональными эквивалентами при миграции от Windows к Linux. Заметьте, что в графе приложений, работающих под Linux, указано одно, наиболее предпочтительное на наш взгляд, хотя почти во всех случаях существует выбор межу несколькими кандидатами.

7.1.5 Описание сетевой инфраструктуры

При описании сетевой инфраструктуры следует обратить внимание на следующие моменты:

- Каков тип сетевой инфраструктуры, к которой подключены клиенты?
- Какова топология сети? Следует описать архитектуру сети, включая все виды локальных и удаленных соединений, пропускную способность различных участков и преобразования протоколов.
- Какие типы протоколов установлены и сконфигурированы на машинах клиентов с целью обеспечения доступа к различным компонентам сетевой инфраструктуры.
- Какие серверы функционируют в сети и какие сервисы обеспечивает каждый из них? В результате ответа на этот вопрос должен возникнуть список серверов, как физических, так и логических, и сервисов, которые каждый из них предоставляет клиентам. К сервисам в этом контексте относятся доступ к файлам, печати, DHCP, к веб-данным, приложениям и т.п.
- К каким базам данных должен быть обеспечен доступ и каким образом? Соответственно, должен возникнуть список всех серверов баз данных и типов доступа к ним, таких как собственные клиенты (например, клиент DB2® UDB от IBM), доступ посредством API (например, именованные каналы), очереди сообщений (например, Websphere MQ от IBM) и приложений, обеспечивающих доступ (например, SAP-клиент).
- Используется ли доступ к мэйнфреймам с данного рабочего места и если да, то каким образом? Доступ к мэйнфреймам различного типа может быть обеспечен как специфическими клиентами (например, Personal Communications), так и через веб-интерфейс (например, Host on Demand или Host Publisher) или подключение через соответствующие API (например, HLLAPI).

7.1.6 Описание функций конечных пользователей

Точное понимание роли и функций конечных пользователей очень важно при планировании замены операционной системы на рабочих станциях. Можно сказать даже, что это — ключевой фактор успеха миграции. Если конечные пользователи в курсе предстоящих перемен, если они представляют себе, как они будут работать в новом программном окружении, и, особенно, если удается активно вовлечь их в планирование миграции и

Таблица 7.2: Выбор средств для системы X-Window

Используемое приложение	#	Функциональный аналог под Linux
IBM Personal Communications	7	Существует Linux-версия
Intel PRO Ethernet Adapter and Software	7	Существует Linux-версия
Internet Explorer	7	Mozilla
Visual Basic 5.0	7	?
AT&T Network Client	6	Существует Linux-версия
Sametime Client v3.0	6	IBM Community Tools
Lotus Notes 6.0.2	5	Lotus iNotes
Lotus SmartSuite — English	5	OpenOffice
Microsoft Office XP Standard	5	OpenOffice
Adobe Acrobat 5.0	3	Существует Linux-версия
DVDExpress	3	Xine
EasySync Pro	3	Evolution with Pilot-Link
RealOne Player	3	Существует Linux-версия
Shockwave	3	Существует Linux-версия
Windows Media Player 7.1	3	Xine
ADSM v3.1.0.7	2	TSM
Microsoft Visio Viewer 2002	2	Terminal Server
Adaptec DirectCD	1	xcdroast
Adaptec Easy CD Creator 4	1	xcdroast
Adaptec UDF Reade	1	xcdroast
Adobe Acrobat 6.0 Standard	1	Существует Linux-версия
Adobe Photoshop® 7.0	1	gimp
Citrix ICA Client	1	Citrix под Linux
Java 2 Runtime Environment, SE v1.4.2_04	1	Существует Linux-версия
Macromedia Shockwave Player	1	Существует Linux-версия
Microsoft .NET Framework (English)	1	Mono
Microsoft Office 2000 Standard	1	OpenOffice
Nero 6 Ultra Edition	1	xcdroast
NeroVision Express 2	1	xcdroast
Netscape 6 (6.2.3)	1	Mozilla
Norton Speed Disk 7.0 for Windows NT	1	defrag
WinZip 8.0	1	Zip
Yahoo! Messenger	1	Существует Linux-версия

выработку решений, количество возможных организационных неувязок и технических проблем при миграции резко снижается.

Вот примерный список вопросов, которые полезно задать:

- Каковы основные функции, выполняемые данным пользователем или группой пользователей?
- Существует ли распределение пользователей «по ролям»? Если существует, то каково определение каждой роли?
- Существуют ли и для каких групп пользователей набор правил, в соответствии с которыми они должны действовать?
- Существуют ли исключения из этих правил и какие?
- Существуют ли специфические пользовательские настройки, которые необходимо перенести в новую систему (например, список ссылок для броузера или фоновые изображения)?
- Требуют ли текущие приложения для своей работы плагинов или cookies?
- Участвуют ли пользователи в выборе программных средств и если участвуют, то как?
- Есть ли у пользователей возможность самостоятельно устанавливать программное обеспечение, разме-

щенное в общем хранилище?

- Существует ли определенная процедура запроса, установки и удаления программного обеспечения?
- Как часто пользователи обращаются в службу технической поддержки, и каковы стандартные параметры предоставления помощи, такие как время отклика, реакция на повторяющиеся запросы, уровень поддержки и т.д.
- Каков уровень подготовки конечных пользователей, имея в виду базовые функции операционной системы?
- Каков уровень подготовки конечных пользователей, использующих более сложные средства, например, офисные приложения?
- Каков уровень подготовки конечных пользователей, работающих с бизнес-приложениями?
- Какие типы внешних устройств (локальных и удаленных) непосредственно используются на данном рабочем месте?

7.2 Взаимодействие с имеющимися сетевыми службами

Этот раздел посвящен вопросам интеграции Linux-клиента в существующую сетевую инфраструктуру. Естественно, мы рассматриваем тот случай, когда сетевая инфраструктура базируется главным образом на продуктах Microsoft Windows.

7.2.1 Различные типы сетевого окружения

Редко можно столкнуться с ситуацией, когда клиент является первой машиной, переходящей под управление Linux, если мы имеем дело с сетью, состоящей из множества машин, работающих под управлением другой операционной системы. Как правило, миграция начинается с серверов того или иного уровня. В подобных случаях надо описывать инкорпорацию Linux-клиента в сеть, в которой на серверах запущена Samba под управлением Linux. Это сравнительно простая ситуация, но мы в этой книге описываем сценарии миграции именно клиентских машин, поэтому будем рассматривать редкий и более сложный случай, будем считать, что в сети большинство серверов работают не под управлением Linux.

В процессе миграции Linux-клиент может оказаться в следующих вариантах сетевого окружения:

- Домен NT4 с Microsoft Windows NT4 PDC (и, возможно, BDC)
- Домен Active Directory с серверами Microsoft Windows 200x AD
- Домен NT4 с Linux/Samba PDC и BDC
- Другие домены, не базирующиеся на продуктах Microsoft Windows

В большинстве случаев мы имеем дело с первым или вторым вариантом. Третий вариант довольно широко распространен, но в этом случае миграция клиентов обычно следует за переходом сервера на обслуживание домена посредством Samba.

Итак, сосредоточимся на двух первых типах окружения (то есть, на чисто Windows-окружении), на это есть по крайней мере две причины:

- Довольно много доменов соответствуют этой модели
- Количество документов, описывающих подключение Linux-клиента к домену Samba и так огромно.

ПОДСКАЗКА

Если существует план перевода домена на Samba или AD, следует принять это во внимание, планируя миграцию клиентов. Не стоит дважды переделывать одно и то же.

Samba является самым распространенным инструментом интеграции Linux-клиента в сети Windows. Изначально Samba была UNIX-приложением. Samba обеспечивает обмен данными в сети на основе протокола SMB (Server Message Block). Операционная система Microsoft Windows использует тот же протокол для обеспечения доступа к разделяемым файлам и принтерам. Благодаря этому Linux-клиент, на котором запущены компоненты Samba, интегрируется без каких-либо проблем в домен Windows и получает доступ к общим файлам, службам печати, просмотру сетевых каталогов и т.п.

Примеры использования и настройки Samba можно найти в Гл. 11. Здесь мы только сформулировали технические вопросы, которые следует принять во внимание, планируя миграцию.

7.2.2 Аутентификация в домене Windows

В этом разделе мы будем рассматривать технические вопросы, связанные с аутентификацией Linux-клиента в домене Windows, в этом случае пользователи входят в Linux под учетными записями, которые хранятся на PDC.

Аутентификация Linux-клиента в домене Windows нужна по нескольким причинам:

- Linux-клиент должен иметь доступ к сетевым ресурсам, требующим авторизации (сетевые файл-серверы, принтеры и т.п.)
- Пользователям не нужен будет дополнительный пароль для входа в сеть, они будут пользоваться одной комбинацией «имя/пароль»
- Администратору надо будет управлять только одним набором свойств пользователей

Планируя аутентификацию Linux-клиентов в Windows-домене, надо рассмотреть следующие вопросы:

- Для каждого ли клиента будет создан отдельный вход в Windows-домен, или можно использовать утилиту winbind с целью унификации стартового окружения? (Вы можете настроить winbind так, что вход пользователя в Linux-клиент будет автоматически вызывать авторизацию в Windows-домене. В результате этого Linux-клиент становится полноправным членом Windows-домена.)
- Утилита winbind требует аккуратной настойки, есть некоторые особенности, связанные с разделителями.
- Будем ли мы входить в домен NT4 или в домен Active Directory? В последнем случае требуется также использование Kerberos.

Выбор варианта входа для пользователя локально на клиентской машине нехорош тем, что его следствием является масса излишнего администрирования. В этом случае при добавлении нового пользователя в домен нужно будет создавать учетную запись на каждом Linux-клиенте, который теоретически может использовать данный пользователь для входа в сеть. Даже учитывая, что этот процесс можно автоматизировать, ясно, что использование winbind намного удобнее.

Используя winbind надо выбрать разумный winbind-разделитель, а именно, символ, который будет отделять имя домена от имени пользователя, поскольку при использовании winbind к учетной записи добавляется еще наименование домена. Например, пользователь Administrator в домене AD6380 под Linux будет иметь имя AD6380+Administrator, в этом случае в качестве winbind-разделителя выбран знак плюс. Следует проверить, не вызовет ли каких-либо проблем в работе приложений появление знака плюс (+) в имени пользователя. Выбор знака плюс в качестве разделителя является лучшим вариантом в большинстве случаев.

> ПОДСКАЗКА
>
> Как следует протестируйте winbind и winbind-разделитель, убедитесь в правильности настроек до того, как начинать миграцию.

Если аутентификация происходит в домене Active Directory, то кроме Samba должен быть настроен также и Kerberos.

Итак, аутентификацию Linux-клиента в домен Windows обеспечит следующий набор приложений: Samba, Kerberos и winbind.

7.2.3 Использование средств домена для совместного доступа к файлам

Поскольку мы рассматриваем присоединение Linux-клиента к домену Windows и можем предполагать, что значительная часть клиентских машин все еще работает под управлением Windows, то наиболее разумным способом обеспечить совместный доступ в файлам будет использование средств Windows.

Монтирование общих каталогов на Linux-клиентах с использованием **smbmount** или **mount -t cifs** обеспечивает пользователям других файловых систем свободный доступ к ресурсам. Однако, если ресурс открыт не для всех, то для доступа к нему потребуется имя пользователя и пароль, аналогично тому, как это происходит под Windows. Если ресурс монтируется вручную, то в этом нет проблемы, имя и пароль надо просто ввести. Однако, заметим что:

- Монтирование ресурсов вручную с использованием команды mount может быть неудобно для пользователей. Работая под Windows они привыкли, что все ресурсы монтируются автоматически при входе пользователя в систему.
- Монтирование ресурсов вручную вовсе не является обязательным, существуют пути автоматизировать процедуру и выполнить монтирование при входе в систему, подобно тому, как это делается под Windows. Автоматическое монтирование SMB-разделов можно организовать, включив соответствующие строки в файл `/etc/fstab`.

Простым решением будет включить монтирование общих ресурсов в профиль, который активизируется при входе в систему. Но при этом во время регистрации нужно будет вводить дополнительный пароль.

Существует PAM-модуль (Pluggable Authentication Module) pam_mount, который обеспечивает автоматическое монтирование при регистрации. Включая этот модуль в планы миграции, помните, что он еще недостаточно хорошо проработан. Однако других средств автоматического подключения общих ресурсов пока не существует.

Выделите дополнительное время, чтобы всесторонне протестировать работу модуля pam_mount на Linux-клиентах, намеченных к миграции. Если все будет работать хорошо, автоматическое монтирование общих ресурсов значительно облегчит вам жизнь.

ПОДСКАЗКА

Тщательно протестируйте работу модуля pam_mount, запланируйте время на эту работу.

7.2.3.1 Домашние каталоги и разделяемые ресурсы

При планировании миграции встает вопрос о том, где разместить домашние каталоги пользователей. Весьма соблазнительной кажется возможность создать домашние каталоги в общедоступном разделе, работающем под управлением Windows. Это позволит сделать клиентов более «тонкими».

Если домашний каталог размещен на разделяемом пространстве, он становится доступен при входе пользователя с любого Linux-клиента, при этом все настройки пользователя остаются одинаковыми. Если используется графический вход в систему, пользователь получает привычный рабочий стол и окружение.

Однако, следует соблюдать осторожность. Некоторые графические системы (главным образом, KDE) создают символьные ссылки или сокеты в домашнем каталоге пользователя. Между тем, одним из известных недостатков файловых систем SMB, работающих под управлением Samba, является невозможность создавать символьные ссылки и сокеты в рамках этих файловых систем. Это означает, что приложения KDE не смогут корректно работать при удаленном монтировании домашних каталогов на ресурсах SMB.

ПОДСКАЗКА

Использование разделяемых ресурсов для домашних каталогов требует тщательного планирования и тестирования.

7.2.4 Использование сетевых принтеров

Конечно, вполне возможно подключить дополнительные принтеры локально к Linux-клиентам. Однако, это может вызвать неудобства и дополнительные организационные сложности в том случае, если ранее использовались сетевые принтеры. Практически все дистрибутивы Linux сейчас включают систему печати CUPS (Common Unix Printing System), позволяющую при условии, что на клиентской машине установлена Samba, организовать печать на доступные по сети принтеры.

Если планируется, что Linux-клиент будет использовать CUPS для выхода на сетевые принтеры, подключенные внутри Windows-домена, полезно будет рассмотреть следующие вопросы:

- Корректно ли взаимодействуют CUPS и Samba?
- Требует ли доступ к принтеру авторизации? Если да, то пароль должен быть доступен.
- Каковы преимущества использования сервера печати по сравнению с прямой печатью через сетевой интерфейс?
- Поддерживаются ли системой CUPS ваши модели принтеров?

7.2.4.1 Взаимодействие CUPS и Samba

В большинстве дистрибутивов Linux CUPS и Samba взаимодействуют без каких-либо проблем. Тем не менее, это необходимо проверить и, соответственно, при планировании отвести время на проверку.

Наиболее важный момент такой проверки — убедиться, что в CUPS включена поддержка Samba. Информацию о том, как это проверить, можно найти в Разд. 11.7.

7.2.4.2 Принтеры и аутентификация

Некоторые сетевые принтеры могут требовать для своего использования авторизации. Используя CUPS, мы имеем возможность предоставить логин и пароль, для этого они должны быть прописаны в URI принтера. В результате логин и пароль будут записаны в нескольких конфигурационных файлах CUPS.

Планируя доступ к сетевым принтерам, надо принимать в расчет то обстоятельство, что пароль в такой ситуации становится сравнительно легко доступен. Это можно обойти, если сделать доступ к принтеру неавторизованным, то есть сделать принтер доступным для любого пользователя. Либо можно создать специального пользователя для каждого принтера и открыть доступ только для тех клиентских машин, с которых планируется печатать на данный принтер.

7.2.4.3 Принт-сервер или прямая печать?

Существуют два способа использования сетевого принтера: печать через специальный сервер печати или прямая печать через сетевой интерфейс (если такая возможность существует). Если большинство клиентов в домене печатают посредством специального сервера, то лучше будет настроить и Linux-клиент на печать через сервер печати.

Печать через принт-сервер дает известные преимущества:

- Печать однотипна. Нет разницы между Windows-печатью и Linux-печатью.
- Упрощается администрирование. В том случае, если требуется отключить или заменить принтер, все изменения делаются только на сервере печати и не затрагивают клиентских машин.

7.2.4.4 Поддерживает ли CUPS модель/драйвер вашего принтера?

Дистрибутивы Linux могут существенно различаться по списку поддерживаемых принтеров и драйверов. Следует убедиться, что все нужные модели принтеров поддерживаются выбранным дистрибутивом.

Если у вас есть старая модель принтера, которая не слишком производительна, то следует подумать, стоит ли заниматься ее адаптацией. Возможно, дешевле будет заменить его на новую модель, которая не потребует никаких специальных усилий для подключения.

7.2.5 DHCP и DNS-конфигурация

В большинстве случаев DHCP и DNS домена Windows будут работать безо всяких проблем и с Linux-клиентом. Оба протокола надо один раз правильно настроить, после чего они будут работать, не создавая никаких сложностей для пользователей.

Однако следует соблюдать некоторую осторожность при совместном использовании DHCP и X11. Linux-клиент обычно принимает имя хоста от DHCP. Если X11 стартует до этого, и пользователь зарегистрирован в системе, при запуске графических приложений могут возникнуть проблемы с Xauth.

> ПОДСКАЗКА
>
> Убедитесь, что имя хоста от DHCP принято Linux-клиентом до старта X11, или используйте графический вход в систему.

7.2.6 Доступ через прокси-сервер

Протокол, который используется для выхода в сеть через прокси-сервер, не зависит от операционной системы. Для того чтобы договорится с прокси-сервером под Windows, достаточно установить корректные параметры веб-броузера.

Таким образом, доступ в сеть с Linux-клиента через прокси-сервер — это простейшая задача в процессе миграции.

7.3 Стандартизация рабочего стола

В этом разделе мы будем рассматривать вопросы, связанные со стандартизацией рабочего стола в Linux. Мы обсудим особенности графической рабочей среды в Linux в общем, а также различные варианты ее реализации, визуальные средства корпоративной идентификации, настройку тем, меню приложений и средства, повышающие удобство работы.

В отличие от классической ситуации в Windows Linux оставляет выбор средств организации графического интерфейса пользователя на всех уровнях. Выбор окружения начинается с выбора наиболее подходящего по своим свойствам дистрибутива. Вторым шагом является выбор менеджера рабочего стола, затем можно заняться более тонкой настройкой внешнего вида и функциональности.

Близкой к теме стандартизации рабочего стола является тема персональных настроек клиента. В данном контексте под термином персональные настройки понимается вся совокупность настроек внешнего вида и функциональности, которые отвечают индивидуальному вкусу конечного пользователя. Подробнее об этом Прил. В.

ПОДСКАЗКА

Если для миграции на Linux вы решили использовать дистрибутив, приобретенный у производителя, вы можете использовать входящее в дистрибутив окружение рабочего стола в качестве базового. В этом случае у вас не возникнет необходимости разрабатывать свой стандарт рабочего стола с нуля, как обсуждается в этом разделе. Так что при выборе промышленного дистрибутива важным критерием отбора является соответствие входящей в дистрибутив по умолчанию графической среды функциональным потребностям клиента.

7.3.1 Дистрибутивы Linux

Поскольку сама ОС Linux и основная часть сопутствующих программ являются программами с открытым кодом, практически любая организация может собрать и выпустить в свет свой дистрибутив. Существуют сотни дистрибутивов, каждый из них занимает свою нишу. Некоторые создавались с целью упростить установку и использование Linux для неподготовленных пользователей, другие были ориентированы на промышленное использование, существуют специальные дистрибутивы для тех, кому важна максимальная свобода в перенастройке системы. В этом издании мы уделяем внимание главным образом двум дистрибутивам: Red Hat Enterprise Linux и Novell Linux Desktop (который базируется на SUSE Linux фирмы Novell).

При выборе дистрибутива имеет смысл принять во внимание следующие факторы:

- Наличие поддержки и ее стоимость
- Набор приложений, включенных в дистрибутив
- Средства администрирования, включенные в дистрибутив
- Наличие программного обеспечения независимых производителей, ориентированного на этот дистрибутив

7.3.2 Менеджеры рабочего стола Linux

Существует множество средств управления графическим рабочим столом под Linux (см. также Разд. 8.2.4), но наиболее распространенные дистрибутивы в качестве основного средства управления рабочим столом используют KDE и GNOME, оба эти менеджера предоставляют пользователям полнофункциональное рабочее окружение. Они содержат не только графический инструментарий и средства управления окнами, но и средства работы с иконками, меню панели, включают такие приложения как броузеры, средства записи компакт-дисков и стандартные офисные программы. Для этих графических сред написано множество дополнительных программ, таких как полноценные офисные пакеты, мультимедиа проигрыватели, средства просмотра и редакторы изображений, универсальные почтовые клиенты и средства работы в группе. Все приложения, написанные для каждой из этих сред, имеют сходный внешний вид и используют общие глобальные настройки. Однако настройки, сделанные для одного рабочего стола, не окажут влияния на приложения, ориентированные на другой рабочий стол. Например, если выбрать определенную цветовую гамму для GNOME, то GNOME-ориентированный почтовый клиент Evolution будет использовать эту гамму, а KDE-ориентированный клиент Kontact — нет. Поэтому при выборе программного обеспечения предпочтение отдается обычно средствам, ориентированным на основной рабочий стол.

ЗАМЕЧАНИЕ

Приложения, написанные для KDE, будут работать в среде GNOME и наоборот, но их внешний вид и поведение могут не соответствовать текущему окружению рабочего стола.

Тесты показывают, что современные менеджеры рабочего стола, такие как GNOME или KDE, нисколько не уступают в простоте освоения и использования рабочему столу Windows. Пользователи, которые раньше не имели опыта работы с интерфейсом Windows, легко осваиваются с рабочим столом под управлением Linux. Тем пользователям, которые долго работали с графическим окружением Windows, приходится потратить некоторое время на ознакомление с новыми возможностями, но обычно переобучение происходит достаточно быстро. Полезную информацию об удобстве и простоте использования графических средств Linux можно найти в сети по адресам:

http://www.betterdesktop.org

http://www.openusability.org

Большая работа в различных областях была проделана для достижения простоты в использовании и универсальности рабочего окружения:

- Специальные возможности (средства для работы за компьютером для слепых или людей с ограниченными физическими возможностями)
  - экранная лупа
  - экранная клавиатура
  - клавиатуры с расширенными возможностями
  - программы чтения с экрана
  - Брайль-мониторы
  - программы генерации речи из текста
  - системы распознавания речи
- Стандарты для разработчиков приложений
  - Human Interface Guidelines (HUG) для GNOME:

    http://developer.gnome.org/projects/gup/hig/
  - User Interface Guidelines для KDE:

    http://sourceforge.net/projects/partima
- Интернационализация (поддержка UTF-8, Pango)

7.3.2.1 Менеджер рабочего стола KDE

Менеджер рабочего стола KDE (http://www.kde.org) обеспечивает клиентам, работающим под управлением Linux и других UNIX-подобных операционных систем, все функции по обслуживанию рабочего стола и запуску приложений в стиле, близком к Windows. KDE написан на основе кросс-платформенного инструментария Qt фирмы Trolltech, с использованием компонентной модели KParts и механизмов DCOP IPC/RCP, которые будут обсуждаться в Прил. С.

Используя KDE можно располагать на рабочем столе иконки и папки, запускать приложения с панели инструментов KDE, на панель можно вывести дополнительные апплеты (время, состояние батареи или статус сетевого соединения и т.п.). Полезным средством для дополнительной настройки KDE является SuperKaramba:

http://netdragon.sourceforge.net

Дополнительную информацию о KDE можно найти в сети по адресу:

http://www.kde-apps.org

В базовый набор приложений KDE входят:

- Электронная записная книжка Kontact с программами-календарями Kolab или eGroupware.
- Офисный пакет KOffice (включает KWord, KSpread, KPresenter, Kivio, Karbon, KChart)
- Приложение для записи на CD/DVD K3b
- Приложение для редактирования цифровых фотографий Digikam

- Настольная издательская система Scribus
- Веб-броузер и файловый менеджер Konqueror

> ЗАМЕЧАНИЕ
>
> В веб-броузере Safari фирмы Apple используется рендеринг-процессор KHTML.

7.3.2.2 Менеджер рабочего стола GNOME

GNOME — это менеджер рабочего стола с открытым кодом (http://www.gnome.org), разработанный с использованием GUI-инструментария GTK+, в соответствии со стандартами CORBA (ORBit2). GNOME является стандартным менеджером для дистрибутивов фирмы Red Hat, он также используется некоторыми производителями UNIX-систем, например, корпорацией Sun™. Подробную информацию о приложениях GNOME вы найдете в сети по адресу:

http://www.gnomefiles.org

Ниже приведен список наиболее распространенных приложений GNOME:

- Текстовый процессор AbiWord
- Редактор электронных таблиц Gnumeric
- Клиент электронной почты и программа-календарь Evolution
- Ekiga VOIP и Video Conferencing Too (ранее известный как GnomeMeeting, см.:

  http://www.gnomemeeting.org
- GNUCash
- GTKam (GUI для gphoto2)
- Inkscape редактор для работы с векторной графикой
- Файловый менеджер Nautilus

GNOME, подобно KDE, позволяет располагать на рабочем столе иконки и папки. Приложения можно запускать с панели GNOME. Также на панель можно вывести дополнительные GNOME-апплеты (время, состояние батареи или статус сетевого соединения, выбор виртуального рабочего стола и т.п.). Множество опций и дополнительных мини-приложений, десклетов (desklets), предоставляет пакет GDesklets. См.

http://gdesklets.org

http://en.wikipedia.org/wiki/GDesklets

7.3.3 Внешний вид и функциональность

После того как был выбран менеджер рабочего стола, можно приступить к более тонкой настройке внешнего вида и функциональности рабочего стола. Нужно подобрать иконки для дополнительных приложений, можно заметить и иконки встроенных приложений, для того чтобы выдержать единый стиль, полезно подобрать подходящий состав меню приложений.

7.3.3.1 Иконки на рабочем столе

На стандартном рабочем столе расположены иконки наиболее часто используемых приложений. Этот набор можно пополнить или изменить. Может потребоваться, чтобы логотип компании и другие графические элементы, соответствующие бренду, были частью стандартизованного рабочего стола. Например, вы можете заменить

иконку почтового клиента, если она не соответствует цветовой гамме рабочего стола или вставить фирменный логотип вместо стандартной кнопки запуска главного меню (аналог Windows Start Menu).

Следует помнить, что растровые изображения могут искажаться при изменении разрешающей способности экрана, и поэтому их используют все реже. KDE и GNOME обеспечивают достаточно хорошую поддержку векторной графики (SVG) во всех текущих дистрибутивах. При разработке векторных логотипов и других графических элементов для рабочего стола следует придерживаться стандартов W3, которые можно найти, например, на странице http://www.w3.org/Graphics/SVG/. Векторная графика выглядит лучше, чем растровая, легко масштабируется, стандартные векторные форматы адекватно отображаются всеми веб-броузерами. Вследствие этого SVG-форматы становятся постепенно стандартными для оформления рабочего стола в Linux.

Программы для работы с векторной графикой:

- Inkscape: http://www.inkscape.org
- Skencil: http://www.skencil.org
- Karbon: http://www.koffice.org/karbon
- OpenOffice.org: http://graphics.openoffice.org/svg/svg.htm

7.3.3.2 Темы рабочего стола

Графические элементы и логотипы представляют собой только часть того, что называют темой рабочего стола. Для большинства пользователей кажутся удобными стандартные темы, предоставляемые производителями дистрибутивов (например, Industrial дистрибутива Novell Linux Desktop, BlueCurve от Red Hat или Plastik для KDE).

Разработать новую тему рабочего стола не просто, кроме того, если вы готовите стандартное решение для рабочего стола многих пользователей, надо принимать во внимание, что документация производителя опирается на стандартный для данного дистрибутива рабочий стол, и если вы внесете изменения, могут возникнуть проблемы. Если стандартный вариант все же не устраивает, прежде чем тратить время на собственную разработку, посмотрите на те варианты, которые уже подготовлены и доступны в сети, напрмер, на страницах:

- http://themes.freshmeat.net
- http://www.customize.org
- http://www.kde-look.org
- http://art.gnome.org
- http://www.crystalgnome.org

Даже если вы решили использовать стандартную тему, вы имеете возможность внести множество мелких изменений и усовершенствовать внешний вид рабочего стола.

Некоторые приложения имеют свои собственные подсистемы поддержки тем, например, Mozilla или OpenOffice.org. Есть еще один момент, который надо принять во внимание при выборе темы рабочего стола. Если запускать приложения менеджера GNOME на рабочем столе под управлением KDE, то такие приложения будут использовать настройки темы GNOME, а не настройки темы KDE. Если есть необходимость в запуске таких приложений, то имеет смысл выбрать тему, которая реализована для обоих рабочих столов. Например, стандартная тема BlueCurve фирмы Red Hat реализована в вариантах и для KDE, и для GNOME. Также есть тема Firefox, соответствующая теме Industrial, стандартной для рабочего стола дистрибутивов Novell Linux Desktop. Найти ее можно на странице:

http://primates.ximian.com/~glesage/stuff/firefox/

Пользователи, привыкшие работать в Windows, могут использовать темы Windows XP для рабочих столов под управлением KDE или GNOME, привычный вид экрана поможет быстрее освоиться с другими изменениями.

7.3.3.3 Дизайн меню приложений

Еще одним аспектом стандартизации рабочего стола является выработка разумной структуры и состава меню приложений. Если речь идет о промышленном использовании компьютера, нет смысла давать пользователю на выбор несколько приложений, выполняющих одну и ту же функцию. Фирма Novell в дистрибутиве SUSE Linux

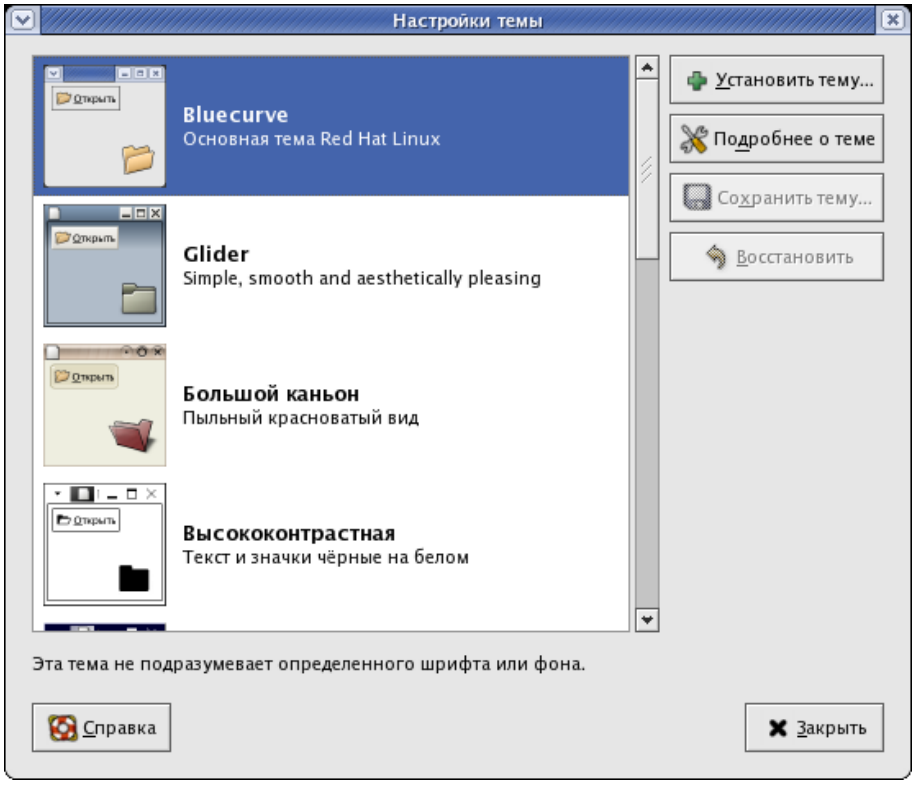

Рис. 7.1: Диалог выбора темы рабочего стола в GNOME

Enterprise Desktop 10 пошла еще дальше и ограничила состав главного меню несколькими пунктами, связанными с основными приложениями, и дополнительно оставила еще несколько точек входа для выполнения некоторых часто повторяющихся действий. Пример экрана со стандартной структурой меню SLED 10 можно видеть ниже.

В некоторых случаях может потребоваться еще больше ограничить выбор возможностей запуска приложений, более того, иногда надо закрыть возможность для пользователя изменять настройки рабочего стола (например, добавлять или изменять апплеты на рабочем столе, добавлять новые кнопки запуска в меню приложений, изменять изображения, расположенные на экране и т.п). Для управления окружением в KDE существует удобный инструмент KDE kiosk. Первоначально он был спроектирован для формирования безопасного рабочего окружения киоск-терминалов, однако с его помощью удобно вводить и более мягкие ограничения для рабочего стола промышленного настольного компьютера. Возможности этого инструмента будут обсуждаться ниже в Разд. 9.1.

Внимание

Изменение прав доступа или владельца конфигурационных файлов не дает гарантии защиты от несанкционированных действий, так как пользователь может переименовать существующие или создать новые файлы с другими полномочиями.

В том случае, если используются оба менеджера, и GNOME, и KDE, предпочтительно бывает унифицировать внешний вид рабочего стола и ограничения на запуск приложений. Это не простая задача, так как до сих пор у GNOME и KDE используются конфигурационные файлы различных форматов. Однако сейчас оба менеджера, KDE версии 3.3 и GNOME версии 2.12, начали поддерживать стандарт структуры меню приложений Desktop Menu Specification, разработанный группой freedesktop.org. Описание этого стандарта можно найти в сети по адресу:

http://standards.freedesktop.org/menu-spec/latest

Рис. 7.2: Упрощенное, ориентированное на решение конкретных задач меню (дистрибутив SUSE Linux Enterprise Desktop 10)

7.3.4 Ограничения для пользователей

Различные пользователи настольного компьютера должны иметь различные уровни доступа и различные возможности по управлению рабочим окружением. Если пользователь является техническим специалистом, ему разумно дать полный доступ к контролю над рабочим окружением, включая пароль суперпользователя, другим же пользователям доступ надо ограничить.

В дистрибутивах Linux доступ пользователей к возможностям программного и аппаратного обеспечения определяется принадлежностью пользователя к той или иной группе. Например, может существовать группа audio для тех пользователей, которым разрешен доступ к звуковой карте, и группа games, при этом запуск всех установленных в данной системе игр будет разрешен только для пользователей, относящихся к этой группе.

7.3.5 Выбор приложений

Одним из основных достоинств систем с открытым кодом является широкая возможность выбора приложений. Однако в некоторых ситуациях возможность выбора нежелательна. Например, перед вами дистрибутив, в который включены четыре текстовых редактора, пять терминальный программ и шесть почтовых клиентов. Если вы оставите в системе все эти приложения, пользователь, для которого Linux является новой рабочей средой, может запутаться. Число установленных приложений должно варьироваться в зависимости от возможностей и потребностей работников. Для базовой конфигурации рабочей станции нужен один почтовый клиент и один веб-броузер, лучше не оставлять пользователям и возможности выбора менеджера рабочего стола. Естественно, это не касается рабочих станций технических специалистов.

7.3.6 Файловые системы и разделы

Далее, в Разд. 8.2.6 подробнее обсуждаются типы файловых систем, используемые в Linux и варианты их физического размещения. В процессе создания базового рабочего окружения надо определить список разделов, которые будут принадлежать стандартной конфигурации. Также должны быть стандартизированы точки монтирования сетевых разделов, кроме прочего, это облегчит и техническую поддержку в дальнейшем. Чтобы новые названия разделов, появившиеся на иконках каталогов, не смущали пользователей, привыкших к однобуквенным обозначениям дисков под Windows, добавьте эти буквы в скобках после имен разделов.

Вам нужно решить, будете ли вы использовать списки прав доступа (ACL) или стандартные для Linux уровни доступа на основе ограничений файловой системы. В большинстве случаев достаточны стандартные средства определения прав доступа. Если все же требуется использование ACL, надо убедиться, что все инструменты, обеспечивающие безопасность в системе, поддерживают ACL.

7.4 Планирование миграции приложений

Для каждого приложения, допускающего миграцию под Linux, необходимо определить способ перевода на новую платформу. Существует несколько возможных сценариев миграции. Например:

- Кросс-платформенные приложения: те приложения, которые работают как под Microsoft Windows, так и под Linux. Эта тема уже обсуждалась в Разд. 6.2.1. Кросс-платформенные приложения существенно упрощают процесс миграции, их следует использовать везде, где это возможно.
- Функциональные аналоги: приложения, обеспечивающие сходную функциональность и импорт данных из форматов Windows-приложений. Например, пакет OpenOffice.org обеспечивает ту же функциональность, что Microsoft Office, и умеет импортировать файлы соответствующего формата. В Разд. 6.2.2 рассматривались функциональные аналоги для архиваторов, файловых навигаторов, мультимедийных средств и т.п.
- Сервер-ориентированные приложения: этот подход может быть полезен для приложений, не имеющих функциональных аналогов в Linux. В этом случае функциональную часть работы приложения стараются перенести на тот или иной вид сервера, а на клиентской машине оставляют только ту часть, которая необходима для обеспечения удаленного доступа и интерфейса пользователя. Такой подход обсуждался в Разд. 6.2.4.
- Эквивалентное приложение, работающее через веб-интерфейс: заменить имеющееся платформо-зависимое приложение на приложение, имеющее кросс-платформенный веб-интерфейс. См. Разд. 6.2.3.

7.4.1 Возвращение к архитектуре клиент/сервер

При планировании миграции приложений на Linux могут возникнуть весьма серьезные причины для возвращения к технологии обработки данных типа клиент/сервер для некоторых приложений. Зачастую оказывается рациональным перейти от локального приложения к приложению, работающему через веб-сервисы, независимо от смены операционной системы. Тем более при планировании миграции надо обратить внимание на такую возможность.

Ниже мы рассмотрим подробнее логическое разделение рабочих станций в соответствии с «весом» приложений, которые на них запускаются, это поможет выделить кандидатов на включение в архитектуру клиент/сервер.

7.4.2 Разделение клиентов «по весу» — Тонкий, Стройный и Толстый

Может быть главным сдвигом при переходе от одной операционной системы на клиентских машинах (Windows) к другой (Linux) является изменение структуры распределения «веса» рабочих станций, которые используются в организации.

Для того чтобы сделать дальнейшее обсуждение более конкретным, выделим три логических типа клиентов на рабочих станциях:

- Тонкий (Thin): всегда подключен к сети, нет собственных установленных локально приложений, кроме тех, которые обеспечивают интерфейс пользователя; таким приложением может быть веб-броузер или другой подобный сервис.
- Стройный (Slim): как правило, подключен к сети; некоторые приложения запускаются локально, установлены (если необходимо) средства синхронизации с сервером; могут быть установлены оригинальные клиент/сервер приложения.
- Толстый (Fat): как правило, работает автономно; большинство используемых приложений установлено и запускаются локально.

На рисунке ниже показано различие в схемах расположения логических частей приложений на клиентской и серверной частях для различных «по весу» клиентов в системе клиент/сервер.

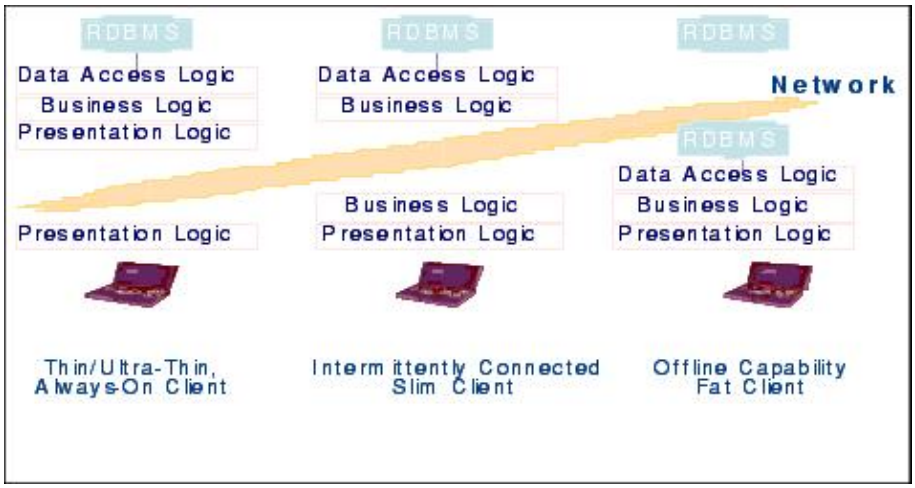

Рис. 7.3: Расположение логических частей приложений для различных типов клиентов

7.4.3 Системные решения для Linux

Следует заметить, что в последнее время независимые производители ПО заметно активизировались в деле разработки приложений для Linux-клиентов, список ПО, работающего под Linux, растет с каждым днем.

Многие производители разрабатывают версии своих продуктов, работающие под Linux, другие переходят на веб-ориентированные модели, реализация которых базируется на Java, клиенты таких приложений работают под Linux без каких-либо проблем, при условии, что в них не использованы какие-либо специфические возможности, например, ActiveX.

Практически каждый день появляются новые приложения под Linux, как общего назначения, так и специализированные. Если бы мы попытались привести здесь даже частичный список, это заняло бы слишком много места. Для тех, кто интересуется этим вопросом, укажем полезный интернет-ресурс, страничку IBM, содержащую список приложений, работающих под Linux:

http://www.ibm.com/linux/solutions

7.5 Планирование управления Linux-клиентом

В этом разделе обсуждаются некоторые методы эффективного управления Linux-клиентом. Если корпоративная сеть содержит около 20 клиентов, то уже в таком случае индивидуальная установка и администрирование каждого клиента становятся непрактичными. Дополнительным преимуществом Linux является возможность удаленного доступа для администрирования клиента посредством соединения через ssh, telnet или xterm.

Здесь обсуждаются вопросы администрирования уже установленного Linux-клиента, возникающие после миграции. Мы не будем обсуждать вопросы установки или удаления Linux-клиентов. Вопросы текущего администрирования системы являются существенным при планировании миграции, так как их изучение позволяет избежать проблем при управлении клиентами после миграции.

В конце раздела мы более подробно рассмотрим два промышленных дистрибутива Linux, чтобы продемонстрировать, какими средствами решаются задачи управления Linux-клиентами в каждом из этих продуктов.

ПОДСКАЗКА

Процедуры ежедневного администрирования достаточно важны при планировании миграции. Тщательное рассмотрение вопросов управления Linux-клиентом позволяет избежать сложностей и коллизий после окончания процесса миграции.

7.5.1 Обновления и изменения операционной системы

Производители дистрибутивов и сообщество разработчиков программного обеспечения с открытым кодом постоянно вводят в использование обновленные версии программных продуктов и «заплатки», исправляющие обнаруженные в процессе эксплуатации ошибки. И с точки зрения увеличения производительности, и с точки зрения безопасности системы необходимо вносить соответствующие изменения в ОС на клиентских машинах. Методы управления процессом обновления включают в себя:

- Автоматизированное обновление ОС и наложение патчей
- Процедуры, помогающие облегчить процесс обновления ОС для конечного пользователя
- Принудительный ввод критических обновлений ОС или патчей в случае необходимости

При автоматизированном процессе обновления ОС изменения вносятся регулярно, никаких действий со стороны конечного пользователя не требуется, он может и не знать о том, что программное обеспечение на его компьютере обновлено. Все пользовательские компьютеры имеют одинаковую версию ОС.

Если принята стратегия обновления силами конечных пользователей, то они становятся активной частью процесса. При таком подходе к регулярному обновлению патчи обыкновенно выкладываются в общедоступный каталог корпоративной сети, а конечные пользователи получают уведомление о том, что они должны скачать и установить их. Такая стратегия может привести к ситуации, когда у части пользователей все обновления будут установлены, а у другой части — нет.

При обоих вариантах поддержания ОС в актуальном состоянии должен работать механизм принудительной установки критических обновлений и патчей. Как правило, для этого администратор запускает централизовано некоторый скрипт, в котором содержится информация обо всех клиентских машинах, на которых надо выполнить обновление. Если принята схема автоматического обновления, то аварийные действия обычно состоят в том, что обновление происходит немедленно, а не по обычному расписанию (например, раз в неделю). Либо скрипт должен содержать средства запуска процесса скачивания и установки. Как правило, необходимость в немедленной установке критических обновлений возникает по соображениям поддержания безопасности системы.

Стандартный дистрибутив Linux состоит из множества частей, эти части обновляются независимо друг от друга. Это означает, что каждая клиентская машина может иметь состояние ОС, отличное от состояния другой машины. Под Windows ситуация другая, там изменения происходят скачкообразно, множество изменений и патчей собираются вместе и выпускаются в форме сервис-пака, их можно применить только одновременно. Для того чтобы эффективно управлять большим числом Linux-клиентов, важно поддерживать их всех на одном уровне, иметь на всех машинах одинаковые версии ОС. Поэтому автоматизированная процедура обновления предпочтительнее того варианта, когда за часть работ несут ответственность конечные пользователи.

Большинство дистрибутивов Linux содержат в себе средства автоматизированного обновления. Например:

- В дистрибутив Red Hat включены инструменты Red Hat Network (up2date); подробнее см. Разд. 7.5.7.1
- В дистрибутиве Suse есть YoU (YaST online Update)

  см.: http://www.novell.com/coolsolutions/feature/11823.html
- В дистрибутив Novell включен ZENworks Linux Management; см. Разд. 7.5.7.2

Из автоматизированных средств обновления ОС с открытым кодом следует упомянуть приложения yum и apt.

ПОДСКАЗКА

Планируйте использование автоматизированной процедуры обновления ОС, чтобы поддерживать одинаковые версии на всех клиентские машинах.

7.5.2 Обновление приложений

Приложения, установленные на клиентских машинах, могут быть частью дистрибутива, тогда их обновление можно осуществлять теми же средствами, какими обновляется сама ОС и которые обсуждались в предыдущем разделе.

Есть несколько подходов к процедуре обновления для тех приложений, которые не являются частью дистрибутива, а именно:

- Использовать скрипт, который скачивает обновленные файлы программ и конфигурации с определенного сервера в сети.
- ПО независимых производителей часто включает в себя программу или скрипт, обеспечивающие процедуру обновления.
- Можно создать пакет, который будет обновляться под управлением apt или yum.
- Если изменения существенные, а клиент тонкий, то разумным подходом будет полностью заменить его на новую версию.

На ранних стадиях при тестировании и пилотной миграции большинство изменений в ПО, как правило, вносится вручную. Однако по мере возрастания количества Linux-клиентов такой подход перестает оправдывать себя, и возникает необходимость включить процедуры автоматизации обновления ПО.

ПОДСКАЗКА

Планируйте создание механизмов автоматизированного обновления ПО на ранних стадиях тестирования Linux-клиента или во время пилотной миграции.

Обратите внимание на то, что параллельно с обновлением приложений, а также с удалением устаревших, должны производится соответствующие изменения в рабочем окружении клиента. Пункты меню и иконки на рабочем столе должны указывать на работающие приложения и ссылаться на актуальную версию ПО в тех случаях, когда несколько версий сосуществует одновременно.

ВАЖНО

Всегда включайте обновление рабочего стола и пунктов меню в процедуру обновления ПО. Если этого не сделать, пользователи просто не смогут работать с обновленными приложениями.

7.5.3 Удаленное управление клиентами

Linux-клиент допускает удаленное управление без установки какого-либо дополнительного программного обеспечения. Администратор может открыть сессию, используя один из стандартных механизмов, например, ssh или telnet. Это позволяет администратору проанализировать и исправить все ошибочные ситуации на клиентской машине удаленно, за исключением тех случаев, когда проблемы касаются работы сети, поскольку тогда удаленный доступ может стать невозможным. Администратор может также удаленно запускать скрипт, который будет заниматься мониторингом и собирать данные о состоянии клиентских машин.

Мониторинг клиентских машин позволяет предупредить возникновение проблем, вместо того чтобы заниматься их устранением тогда, когда пользователи уже не могут работать. Обычно отслеживают состояние жестких дисков, памяти, параметры использования CPU и даже работу некоторых приложений. Существуют как коммерческие инструменты для мониторинга, например, IBM Tivoli Software, так и инструменты с открытым кодом, например, Nagios и Big Brother.

7.5.4 Процедура добавления или замены клиентов

После того как значительная часть клиентских машин уже работает под Linux и ситуация становится стабильной, все равно остается задача включения новых клиентов или замены ОС на некоторых работающих настольных компьютерах. Для таких случаев полезно иметь под рукой готовый дистрибутив, в который включены все текущие обновления. Конечно, зачастую приходится делать некоторые изменения на вновь установленных клиентах, но минимизировать эти изменения можно и нужно.

При замене ОС на клиентской машине надо иметь в виду следующее:

- В случае замены толстого клиента персональные данные должны быть переданы со старого замененяемого клиента или восстановлены из резервной копии.
- Конфигурационные файлы должны быть скопированы.
- Убедитесь, что сохранение старого имени системы не будет приводить к дополнительным ненужным проблемам.
- Если же вы переименовали систему, убедитесь, что старое имя убрано из всех конфигурационных файлов на серверной стороне.

> ВАЖНО
>
> Если используется winbind, то не забудьте при замене ОС скопировать idmap-файл старого клиента, это необходимо, чтобы сохранить существенную информацию об uid/gid пользователя.

В дистрибутивах Red Hat winbind использует следующий idmap-файл:

`/var/cache/samba/winbindd_idmap.tdb`

В дистрибутивах Novell/SuSE winbind использует следующий idmap-файл:

`/var/lib/samba/winbindd_idmap.tdb`

7.5.5 Резервные копии клиентских рабочих мест

Процедура резервного копирования на клиентских машинах нужна только в том случае, если на клиентской машине существуют измененные данные. В том случае, если вся текущая работа делается на сервере, достаточно организовать бэкап сервера. При возникновении проблем на клиентской машине достаточно будет переустановить на ней ОС с текущего дистрибутива.

Если на клиентской машине установлены локальные приложения или сохраняются некоторые данные, то необходимо организовать бэкап. Общепринятым вариантом резервного копирования в конфигурации клиент-сервер

является установка на клиентской машине ПО, которое регулярно сохраняет копии локальных данных рабочей станции на специальном бэкап-сервере. Как правило, данные такого сервера дополнительно копируются на статический носитель, например, на магнитную ленту. Примером подобного ПО для Linux может служить IBM Tivoli Storage Manager.

Однако некоторые конфигурационные файлы и временные копии недавно использованных данных на локальных машинах могут изменяться даже при условии, что основные данные для всех приложений хранятся на сервере. Например, пользователь может перенастроить рабочий стол на свой вкус. После восстановления системы со стандартного дистрибутива такие изменения пропадут. Возникает вопрос, стоит ли заниматься сохранением личных настроек и временных файлов. Можно предложить пользователям самим восстанавливать рабочее окружение в том случае, если они используют личные настройки.

Можно составить опись подобных файлов на ранних стадиях миграции и написать скрипт, который будет собирать такие файлы в виде архивов и сохранять их на сервере. Поскольку размер этих файлов невелик, такая процедура не будет мешать работе и обеспечит пользователю дополнительное удобство в случае замены системы.

Если вы принимаете такую стратегию, то не забывайте обновлять список личных файлов, подлежащих архивированию при обновлении версий приложений.

ПОДСКАЗКА

Применяйте архивирование конфигурационных и временных файлов, чтобы иметь возможность восстановить привычное для пользователя рабочее окружение после переустановки системы.

7.5.6 Антивирусные мероприятия

Защиту от заражения вирусами следует включать сразу, как только Linux-клиент запущен в работу. Не надо успокаивать себя тем, что пока известно очень небольшое число Linux-вирусов. По мере распространения рабочих станций под Linux их число может возрасти.

Необходимо ввести три наиболее важных способа защиты от вирусов и червей:

- фильтр, выявляющий вирусы в почтовых сообщениях, установленный на почтовом сервере;
- регулярное сканирование рабочих станций на наличие вирусов;
- брандмауэр, препятствующий проникновению вирусов на клиентские машины по локальной сети.

Существует несколько фильтров для почтовых серверов, как коммерческих, так и основанных на открытых кодах.

Сканировать на наличие вирусов следует все части локальной файловой системы, для этого также существует ряд антивирусов. Следует регулярно обновлять антивирусную базу используемого инструмента, желательно делать это автоматически. Пользователи не должны иметь возможность остановить сканирование своей машины антивирусной программой.

Брандмауэр на Linux-клиентах должен быть сконфигурирован так, чтобы защитить рабочее место от проникновения вирусов и червей по сети. Конфигурация брандмауэра должна быть защищена от изменений конечным пользователям, ее следует регулярно централизованно обновлять.

ПОДСКАЗКА

Планируйте включение антивирусной защиты Linux-клиентов на ранних стадиях миграции. Не успокаивайте себя тем, что пока не известно по-настоящему опасных Linux-вирусов. Они могут появиться.

7.5.7 Службы администрирования в промышленных дистрибутивах

В этом разделе мы рассмотрим, как с помощью встроенных интегрированных сред управления системой осуществляется администрирование в дистрибутивах Red Hat Enterprise Desktop и SUSE Linux Enterprise Desktop.

7.5.7.1 Администрирование в дистрибутиве Red Hat Desktop

Дистрибутив Red Hat Desktop базируется на Red Hat Enterprise Linux. Как и во всех продуктах этой линейки, в этом дистрибутиве для администрирования и обновления версий применяется механизм RHN (Red Hat Network).

Red Hat предлагает три варианта конфигурации соединения с RHN: Хостинг, Прокси, Саттелит (Hosted, Proxy, Satellite). Обычно доступ к обновлениям происходит непосредственно, клиент связывается с одним из RHN серверов Red Hat. Такой вариант называется Хостинг. В дистрибутив включена возможность установки RHN прокси сервера или RHN сателлит сервера локально. RHN прокси сервер кеширует обмен между RHN серверами Red Hat и клиентскими машинами в локальной сети, использование RHN сателлит сервера позволяет иметь локальную копию RHN сервера Red Hat.

Варианты Прокси и Сателлит полезны в том случае, если число рабочих станций в сети достаточно велико, и получение обновлений непосредственно с серверов Red Hat существенно увеличивает трафик. Эти варианты хороши также и тем, что повышают безопасность системы за счет уменьшения времени соединения с Интернетом. Дополнительным преимуществом варианта Сателлит является то, что клиенты в локальной сети могут получать обновления независимо от наличия в текущий момент связи с Интернетом.

Инструментом для обновления версий и наложения «заплаток» в RHN является up2date. Для уведомления пользователей о появлении обновлений или патчей используется специальная утилита оповещения об обновлениях в RHN (RHN alert notification tool). Процесс обновления версий может быть автоматизирован.

Использовать RHN можно с тремя типами модулей:

- Update
- Management
- Provisioning

Выбор того или иного модуля определяет уровень сервиса, который вы будете получать от RHN.

Модуль Update дает базовые возможности в области управления клиентом:

- Доступ к веб-интерфейсу для управления системой.
- Получение по электронной почте уведомлений о появлении свежих версий и об обнаруженных ошибках.
- Вместе со списком обнаруженных ошибок присылается список доступных патчей для различных систем.
- Проверка зависимостей RPM дает уверенность, что будут корректно проведены все связанные между собой обновления.
- Существует процедура автоматизации обновлений, которая загружает нужные данные и автоматически их устанавливает.

Модуль Management дает все возможности модуля Update плюс дополнительные возможности управления процессом:

- Есть возможность создавать группы систем и управлять ими однотипно.

- Системные права доступа могут быть приписаны нескольким администраторам и относиться к группам систем.
- Можно создать расписание обновлений для систем и групп.
- Специальная поисковая система дает возможность осуществлять поиск по пакетам, патчам или системным спецификациям внутри групп или отдельных систем.
- Можно сравнивать пакеты на различных системах или делать дамп конфигурации для последующего сравнения.

Модуль Provisioning особенно полезен в том случае, если сеть RHN используется для создания новых клиентов. В дополнение ко всем возможностям модулей Management и Update вы получаете:

- Bare metal provisioning — инструмент для автоматической установки операционной системы, пакетов и ключей активации. Использует механизм kickstart.
- Existing state provisioning — инструмент, позволяющей настроить текущую систему в соответствии с параметрами, которые заданы системой-шаблоном.
- Функциональность, которая позволяет отменить все внесенные изменения и вернуться к предыдущему состоянию системы.
- Настройка конфигурации может быть соединена с процессом установки системы процедурой kickstart.
- Установки и настройка приложений, представленных в виде пакетов RPM.
- Специальное средство для записи конфигураций для установок типа kickstart.

Все эти опции доступны для варианта связи Хостинг. Если же используется конфигурация Прокси или Сателлит, то добавляются еще и пользовательские каналы, интегрированные сетевые установки и многое другое.

Сеть RHN позволяет также организовать автоматическое управление RHD-клиентами. В конфигурации Хостинг управление ограничено свойствами операционной системы и пакетами, включенными в дистрибутив. Другие конфигурации позволяют создать пользовательские каналы, при помощи которых можно управлять приложениями независимых производителей.

Дополнительную информацию о RHN можно найти в сети по адресу:

http://www.redhat.com/software/rhn

а также в издании *Linux Client Migration Cookbook, Practical Planning and Implementation Guide for Migrating to Desktop Linux*:

http://www.redbooks.ibm.com/redpieces/abstracts/sg246380.html

7.5.7.2 Администрирование в дистрибутиве Novell Linux Desktop

Предпочтительным методом обновления настольной машины, работающей под управлением Novell Linux Desktop, является Novell ZENworks Linux Management. ZENworks Linux Management базируется на Red Carpet Enterprise.

ZENworks представляет собой централизованную систему, посредством которой клиентские машины могут конфигурироваться и откуда они получают свои обновления и патчи. Она располагается за брандмауэром организации и может получать данные для обновления из большого числа внешних источников, включая серверы Red Carpet, серверы обновлений YaST Update и даже от Red Hat Network.

Клиенты подключаются к ZENworks посредством протокола HHTP и получают обновления или патчи при помощи программы rug или red-carpet. Администрирование сервера ZENworks производится либо из командной строки, либо с использованием веб-интерфейса.

Ключ активации, который задается при подключении нового клиента, определяет, какие администраторы будут иметь доступ к клиенту, какие каналы будут доступны и к каким группам будет относиться клиент.

ZENworks Linux Management включает в себя следующие средства:

- Улучшенный ACL — работа с ключами активации. Позволяет ассоциировать администраторов, каналы, группы и типы извещений с ключом активации.

- Контроль отложенных обновлений — администратор может видеть, у каких клиентов отложено обновление версий и насколько важные обновления пропущены.
- Группирование пакетов — программные пакеты можно объединять в группы. Для таких групп поддерживается контроль версий и проверка зависимостей, как для обычного пакета.
- Создание групп «на случай» — администратор может создавать временные группы всегда, когда ему это нужно, например, можно создать группу для всех машин, на которых еще не закончен некоторый процесс обновления версий.
- Улучшенная система транзакций — можно добавлять скрипты для запуска немедленно или в будущем, есть возможность выполнить откат от изменений, сделанных тем или иным скриптом.
- Установка прав доступа для пользователей — конфигурационные файлы клиентских машин можно контролировать с сервера.
- Тестовый запуск транзакции на серверной стороне — можно протестировать подготовленную транзакцию для проверки удовлетворения зависимостей на серверной стороне, не затрагивая работу клиента.
- Сопоставление клиентов — инструмент, который позволяет сравнить состояние двух или более клиентских машин.
- Возможность выбора базы для хранения данных — поддерживается PostgreSQL и Oracle.
- Кеш-сервер — сервер, который собирает последние обновления программного обеспечения с нескольких подсетей.
- Улучшенный инструмент для работы с зеркальными серверами — позволяет генерировать группы пакетов, используя описания, предоставляемые зеркалированными серверами.

При создании новых клиентов имеются дополнительные возможности:

- Поддержка нескольких серверов — клиент может быть ассоциирован с несколькими серверами ZENworks Linux Management.
- Возможность отката к прежней версии — процедура может быть запущена как локально, так и с серверной стороны.
- Расширенная информация о системе — клиент посылает на сервер информацию об аппаратных средствах, программном обеспечении и системной конфигурации.

Дополнительную информацию о Zenworks Linux Management можно найти в сети по адресу:

http://www.novell.com/products/zenworks/linuxmanagement/

а также в издании *Linux Client Migration Cookbook, Practical Planning and Implementation Guide for Migrating to Desktop Linux*:

http://www.redbooks.ibm.com/redpieces/abstracts/sg246380.html

7.6 Сравнение процедуры миграции настольных машин и ноутбуков

Различие между настольными машинами и ноутбуками состоит не только в том, что ноутбуки имеют специфические аппаратные средства, но также и в том, что приемы работы на них могут отличаться от приемов работы на стационарном компьютере. Ноутбук зачастую работает автономно, без подключения к сети и нуждается в синхронизации данных при подключении.

В процессе работы ноутбук может подключаться к различным типам сетей, он должен иметь соответствующие аппаратные средства. Программное обеспечение, установленное на нем, должно уметь работать как онлайн, так и офлайн, программы должны иметь механизм адаптации к текущему типу соединения, если это необходимо.

ПОДСКАЗКА

Поскольку пользователи ноутбуков будут скорее всего относиться к правому сегменту в функциональном разбиении, показанном на Рис. 6.1, миграция системы для них требует дополнительного планирования, как для установления совместимости аппаратных средств, так и для миграции приложений.

7.6.1 Аппаратные средства

Обсуждая поддержку аппаратных средств в Linux, следует прежде всего отметить, что она значительно расширилась в последние годы. Многие производители оборудования, наблюдая распространение использования Linux как на домашних, так и на офисных машинах, стали снабжать свои устройства драйверами, обеспечивающими их работу под Linux. К сожалению, сделано еще далеко не все, особенно много пробелов в части поддержки аппаратных средств беспроводного доступа к сети, есть также специфические устройства типа Win-модемов или GDI-принтеров. Несколько проектов находятся в разработке, среди них можно упомянуть Wlan-драйвер для чипсета Intel Centrino™. Проблему как правило представляют прошитые в ПЗУ программы и файлы, код которых закрыт, в результате чего разработчикам свободного ПО приходится заниматься обратным проектированием интерфейса устройства, чтобы добиться его работоспособности под Linux.

Подобные разработки приводят даже к идеям типа проекта ndiswrapper, применение которого дает возможность загрузки драйверов Windows ndis с целью использования сетевых карт под Linux. Ndis — это API Windows для драйверов сетевых карт, а ndiswrapper позволяет использовать тот же API и обращаться к драйверам, как к модулям ядра Linux. Адрес этого проекта в сети:

http://ndiswrapper.sourceforge.net

Следует оценить целесообразность использования имеющихся аппаратных средств на стадии планирования миграции (см. Разд. 7.1.2). Возможно, устаревшее оборудование потребует очень больших усилий для того, чтобы заставить его работать в новых условиях, и будет целесообразно заметить его на новое, совместимое с Linux.

Это правило особенно применимо при выборе модели ПК. Практически все крупные производители ПК обеспечивают выбор моделей, полностью совместимых с Linux. Информацию о совместимости компьютеров, производимых IBM, можно найти в сети по адресу:

http://www.ibm.com/pc/support/site.wss/MIGR-48NT8D.html

Для настольного компьютера определить, будет ли он работать под Linux, достаточно просто, поскольку все компоненты, включая сетевые и видеокарты, известны и легко заменяются, всегда можно проверить, поддерживается ли именно эта модель. Неясность может возникнуть только тогда, когда карта встроена в материнскую плату, но в этом случае проблема решается добавлением отдельной карты, чипсет которой поддерживается Linux.

В ноутбуке замена или добавление карты невозможны. Поэтому прежде чем покупать ноутбук, необходимо убедиться, что данная модель и все ее устройства будут работать под Linux.

Следует обратить особое внимание на поддержку функций управления энергопотреблением мобильного компьютера. Что касается стандартов APM и ACPI, то надо отметить, что под Linux для них остаются еще нерешенные вопросы. ACPI поддерживается начиная с ядра версии 2.6, и большинство функций этого стандарта может быть использовано, но в связи с тем, что существует большое число различных компонент, некоторые проблемы все еще остаются. Для использования моды «standby» в ноутбуке, например, более стабильным вариантом является использование APM.

Таким образом, становится ясно, что следует уделить самое пристальное внимание аппаратным средствам при планировании миграции, особенно это касается ноутбуков. Мы настоятельно рекомендуем убедиться, что все устройства, включенные в ноутбук, будут работать под Linux. Возможно, потребуется дополнительная работа для того, чтобы в итоге получить подходящий дистрибутив, в котором поддерживаются все ваши аппаратные средства.

7.6.2 Периферийные устройства

Еще в процессе инвентаризации оборудования вы могли обнаружить, что многие рабочие места оснащены не только настольными машинами, но и различными видами периферийных устройств. Примерами таких устройств могут быть:

- Локальные принтеры
- Серверы печати
- Сканеры
- Плоттеры
- Цифровые фотокамеры
- Устройства для чтения/записи карт памяти

Планируя миграцию, важно оценить это периферийное оборудование так же, как и основные аппаратные средства клиентских машин. Создание описи и оценка оборудования подробно обсуждались в разделе Разд. 7.1.2.

Мы рекомендуем сравнить ваш инвентаризационный список со списком протестированных и стандартизированных для работы под Linux устройств, особенно если речь идет о дешевых периферийных устройствах. Возможно будет дешевле купить новое устройство, про которое известно, что оно поддерживается Linux, чем пытаться заставить работать старое.

Информацию о проектах разработки программного обеспечения с открытым кодом для периферии можно найти на страницах в сети по следующим адресам:

- Система поддержки печати CUPS (Common Unix Printing System) (см. Разд. 11.7):

  http://www.cups.org

- Поддержка сканеров SANE (Scanner Access Now Easy):

  http://www.sane-project.org

- Проект разработки драйвера смарт-карт под Unix:

  http://smartcard.sourceforge.net

- Проект gphoto2 — поддержка цифровых фотокамер:

  http://www.gphoto.org

7.6.3 Соединение с интернетом

В наше время, когда интернет распространен повсеместно и постоянно требуется для работы, возможность соединения с интернетом в различных формах становится одной из важнейших функций персонального компьютера. Поскольку интернет базируется на UNIX-серверах, а Linux был изначально создан как вариант UNIX для персональных компьютеров, то очевидно, что при работе Linux-клиента в интернете проблем возникнуть не может.

Что же касается подключения к интернету, то если речь идет о стандартном варианте с использованием сетевых карт Ethernet, то проблем тоже нет. Практически все популярные Ethernet-карты поддерживаются Linux. С беспроводной связью дело обстоит сложнее. Многие сетевые карты для беспроводной связи, которые появились вначале, были ориентированны на сегмент рынка, называемый SOHO (от Small Office/Home Office), то есть на домашние компьютеры и компьютеры для небольшого офиса. В тот момент в этом сегменте было мало Linux-машин, и производители решили сэкономить, не включая драйверы для Linux. Хорошей тактикой при выборе подобных карт будет обратиться к изделиям более крупных и надежных производителей, например, компания Cisco предлагает Linux-драйверы для всех своих беспроводных сетевых карт.

Соединение с интернетом посредством DSL или кабельного модема не представляет сложность, поскольку эти устройства работают через сетевой интерфейс Ethernet. IP-соединение посредством этих устройств устанавливается без проблем при помощи протоколов PPP или PPPoE. Провайдеры услуг для кабельных модемов обычно предоставляют IP-адрес посредством DHCP.

С модемным сеансовым типом соединения иногда могут возникнуть проблемы. Существуют так называемые win-модемы, особенно они распространены в бюджетных ноутбуках. Эти модемы используют ресурсы операционной системы для эмуляции аппаратных действий, что удешевляет сам прибор, но зато такой модем требует специфического программного обеспечения для своей работы. Для некоторых win-модемов существуют драйверы под Linux. Актуальную информацию на эту тему можно найти в сети по адресу

http://linmodems.org

Поддержка win-модемов включена в некоторые дистрибутивы, другие же ее не содержат. В конечном итоге может оказаться проще и дешевле подключить внешний модем, чипсет которого поддерживается Linux, чем тратить массу усилий в попытках заставить win-модем работать.

7.6.4 Работа в автономном режиме

В этом разделе мы обсудим некоторые аспекты работы ноутбука в автономном режиме.

7.6.4.1 Автономная работа почтового клиента

Существует несколько полноценных почтовых клиентов для настольных машин (например, Microsoft Outlook, IBM Lotus Notes и т.д.), которые обеспечивают возможность офлайновой работы с почтой. Планируя миграцию для пользователей ноутбуков, следует учитывать это. Выбор правильного почтового клиента на замену Windows-клиенту представляет собой, возможно, наиболее существенное решение при определении миграционной модели. Необходимость обеспечить аналогичную или, по крайней мере, приемлемую функциональность при работе в офлайновом режиме и является причиной того, что миграция ноутбуков является одной из самых сложных задач при планировании миграции клиентских машин под Linux.

Существуют почтовые программы, работающие под Linux, которые обеспечивают офлайновый режим работы с почтой, например, Mozilla Thunderbird. Thunderbird поддерживает оба почтовых протокола, и POP, и IMAP. Узнать подробнее про Thunderbird можно в сети по адресу:

http://www.mozilla.org

Хотя Thunderbird и может работать офлайновом режиме, но он не обеспечивает тех функций, которые берут на себя почтовые клиенты под Windows, такие как Microsoft Exchange/Outlook и IBM Lotus Domino/Notes, а именно — ведение календаря и составление деловых расписаний. В качестве одного из вариантов решения этой проблемы мы намерены в Гл. 10 продемонстрировать применение Novell Exchange Connector в сочетании с почтовым клиентом Novell Evolution.

7.6.4.2 Доступ к файлам в автономном режиме

Как и любой другой пользователь, пользователь ноутбука создает и модифицирует свои собственные файлы и папки. Как правило, личные файлы хранятся не на локальных машинах, а на серверах, к которым организован быстрый доступ по сети, для которых организовано резервное копирование и другие сервисные операции. Но пользователь ноутбука отличается тем, что должен иметь возможность работать с копиями этих файлов в автономном режиме.

Например, Microsoft предоставляет специальные процедуры автоматической синхронизации между серверами и мобильными клиентами. Один из методов синхронизации использует специальную папку Briefcase, в этой папке пользователь должен держать те файлы, которые он хочет синхронизировать. Другой метод появился с выходом Windows 2000, куда была встроена новая функция, называемая offline files. Этот механизм позволяет синхронизировать содержимое целых каталогов, пользователь может работать с ними автономно, все изменения будут автоматически переданы на сервер при включении компьютера в корпоративную сеть.

В Linux нет прямых аналогов синхронизирующих механизмов типа сервер-клиент, встроенных прямо в операционную систему. Задачу синхронизации данных между клиентом и сервером при автономной работе клиента решают различные пакеты, поставляемые независимыми производителями. Например, Novell предлагает инструмент для синхронизации и совместной работы над файлами iFolder. Для iFolder существует два сервера, один из которых основан на открытых кодах, а другой является коммерческим. Подробную информацию можно найти по адресу:

http://www.novell.com/products/ifolder

Еще один способ решения задачи синхронизации — использование протокола rsync. Эта утилита представляет собой мощное средство синхронизации с открытым кодом, rsync обладает гибкой настройкой и позволяет синхронизировать данные целых каталогов на локальной машине и серверной стороне. Благодаря использованию механизма контрольных сумм пересылаются только различающиеся части файлов, что существенно снижает трафик и дает возможность эффективно работать даже по медленным каналам связи. Страница rsync находится по адресу:

http://samba.anu.edu.au/rsync

Используя rsync вы можете организовать на ноутбуке сервис синхронизации файлов между клиентом и серверной стороной, который будет работать по специальному расписанию или по запросу пользователя. Но вам придется позаботиться об организации пользовательского интерфейса, поскольку прямая работа с протоколом слишком сложна для неподготовленного пользователя, или написать скрипт, который будет запускать синхронизацию по расписанию.

7.7 Приложения, не допускающие переноса под Linux

В этом разделе мы рассмотрим проблемы, связанные с приложениями, для которых нет возможности прямого запуска под Linux. Мы предложим альтернативные варианты запуска таких приложений с Linux-клиента.

Мы исходим из предположения, что число приложений, не допускающих переноса на Linux невелико, и что с ними работает незначительная часть конечных пользователей. В противном случае переход на Linux-клиенты может оказаться нецелесообразным.

7.7.1 Что мешает переносу приложений под Linux

Мы называем приложение недопускающим переноса в том случае, если для него верно одно (или несколько) из следующих утверждений:

- Не существует версии этого приложения или функционального аналога, работающего под Linux.
- Портировать приложение под Linux невозможно.
- Лицензионные соглашения делают перенос приложения под Linux невозможным или очень дорогим.

Если приложение было спроектировано таким образом, что оно не допускает миграции, есть несколько путей для продолжения его использования после перехода на Linux-клиент:

- Возможно, приложение может работать на Windows-сервере, и к нему можно организовать доступ с Linux-клиента, используя такие механизмы, как Terminal Server, Citrix Metaframe, NoMachine NX и т.п.
- Проверьте, возможно ли, с точки зрения стоимости мероприятия, запускать виртуальную Windows-машину VMware на рабочих станциях тех клиентов, которым требуется использование Windows-приложений.
- В том случае, если Windows-приложение запускается сравнительно редко, разумно установить на клиентской машине обе операционные системы и организовать возможность альтернативной загрузки.
- Можно оставить несколько машин работать под Windows, и собрать на этих машинах все приложения, не допускающие миграции, для совместного использования теми сотрудниками, которым они необходимы.

7.7.2 Использование Terminal Server, Citrix Metaframe или NoMachine NX

Если приложение может быть запущено на сервере, и к нему возможен терминальный доступ, то хорошим выходом из положения будет использование Terminal Server, Citrix Metaframe или NoMachine NX. Для того чтобы приложение допускало перенос на центральный сервер, должны выполняться некоторые условия:

- Приложение должно уметь работать в многопользовательском режиме. Из этого условия вытекает несколько требований, среди них:
  - пользовательские настройки должны храниться в нескольких местах;
  - несколько пользователей должны иметь возможность запускать приложение одновременно.

- Лицензия на приложение допускает его использование в многопользовательском режиме.
- Ресурсы, которыми пользуется приложение, должны быть разделяемыми.

Для всех трех указанных вариантов доступа, Terminal Server, Citrix Metaframe и NoMachine NX, существуют Linux-клиенты. Клиент для Terminal Server, rdesktop, представляет собой программу с открытым кодом. Клиент Citrix доступен в том случае, если вы приобретаете пакет Metaframe. Компания NoMachine выпустила клиент NX под лицензией GPL.

7.7.3 Использование VMware

Использование VMware (или другого приложения под Linux с аналогичными функциями) дает возможность создавать виртуальное Windows-окружение для работы тех пользовательских программ, которые нельзя перенести под Linux или на центральный сервер, работающий под Windows.

Виртуальная машина представляет собой полный функциональный аналог Windows-машины, и на ней можно повторить в точности то окружение, которое существовало на Windows-клиенте до миграции. В том числе, виртуальная машина входит в локальную сеть с теми же установками, с которыми она входила, будучи Windows-клиентом. При этом для создания соединения должен быть использован бриджинг (bridging) или NAT.

У такого варианта решения проблемы есть несколько недостатков:

- Для виртуальной машины требуется полная лицензия Windows.
- VMware тоже требует лицензии, что приводит к дополнительным расходам.
- Работа VMware на клиентской машине требует администрирования, то есть, дополнительного расхода времени персонала.
- Требуются дополнительные аппаратные ресурсы, а именно, дополнительная оперативная память и дисковое пространство.
- Требуется дополнительная работа с конечными пользователями.

Для того чтобы определить, стоит ли использовать такой вариант запуска немигрируемых приложений, надо оценить количество возможных пользователей виртуальных Windows-машин и стоимость других вариантов решения проблемы.

7.7.4 Использование возможности альтернативной загрузки

Использование клиента с двойной загрузкой для работы с немигрируемыми приложениями схоже с использованием виртуальной машины VMware. Отличие состоит в том, что в этом случае пользователь не может одновременно использовать и Linux, и Windows.

В этом варианте решения проблемы есть следующие недостатки:

- Для загрузки Windows требуется полная лицензия Windows.
- Работа двух операционных систем на одной машине требует дополнительного администрирования, то есть, дополнительного расхода времени персонала.
- При использовании двух операционных систем возникают некоторые дополнительные сложности, среди них:
  - Будет неясно, какой клиент запущен на машине: Windows-клиент или Linux-клиент, возможно следует назначить им разные сетевые имена и IP-адреса.
  - Конечные пользователи не смогут быстро переключаться между различными задачами, для обеспечения работы двух файловых систем придется организовать отдельные разделы.
- У конечных пользователей может быть негативное отношение к необходимости перезагружать машину в процессе работы.

Вариант альтернативной загрузки имеет некоторые преимущества пред использованием виртуальной машины. Во-первых, не требуется покупать лицензию VMware, и это приводит к сокращению расходов. Не требуется дополнительной оперативной памяти, и ресурсы CPU используются не столь интенсивно, так как одновременно работает только одна операционная система, но дополнительное дисковое пространство требуется по-прежнему. Также как и в случае VMware, надо стремиться к минимизации количества клиентов, пользующихся вариантом альтернативной загрузки. Под Windows должно работать только то, что не может работать иначе, чем под Windows.

7.7.5 Что делать, если ни один из вариантов не подходит

Может случиться так, что для некоторых приложений не будет работать ни один из предложенных вариантов решения проблемы. Если от использования таких приложений невозможно отказаться после перехода клиентских машин на Linux по причинам экономическим, юридическим, сложившейся практики или по каким-либо другим причинам, то надо искать запасной выход.

В качестве одного их вариантов можно предложить запуск подобных приложений под Windows-эмулятором, например использовать Wine. Такой вариант можно рассматривать только как временный и использовать его до тех пор, пока не будет найдено кардинальное решение.

Наиболее экономически выгодным решением будет оставить несколько машин работать под Windows и собрать на этих машинах все приложения, которые не допускают миграции. Такие машины по необходимости будут работать в однопользовательском режиме, так как для тех приложений, которые допускают многопользовательский режим, более разумным будет использование центрального сервера под Windows, как это было описано выше в Разд. 7.7.2. С централизованными клиентами можно работать удаленно, доступ к ним может быть организован с помощью такого инструмента как VNC, который работает с протоколом SSL (Secure Socket Layer). VNC обеспечивает нормальную работу приложения при том, что монитор, клавиатура и мышь подключены к удаленной машине, на которой запущен Linux-клиент. Использование протокола SSL гарантирует безопасную передачу данных.

Следует заметить, что все предложенные варианты плохо работают в случае интенсивного использования немигрируемых приложений. Если число таких приложений велико, то миграция клиентов под Linux становится чрезвычайно сложной задачей.

7.8 Развертывание новых клиентов

После того как создана подходящая конфигурация ПО для типичной рабочей станции, следует продумать процедуру установки этой конфигурации на все остальные настольные машины, на которых запланирована миграция. Схема и методика развертывания новой системы должна быть разработана на стадии планирования. Особенно это касается организаций, в которых работают сотни и даже тысячи клиентских машин. Без продуманной и автоматизированной процедуры установки нового ПО миграция остановится на стадии плотного проекта.

Вот некоторые вопросы, которые надо решить, прежде чем приступать к развертыванию новых клиентов:

- Каков будет метод развертывания новых клиентов: установка с использованием готового образа системы и пост-инсталляционных скриптов или автоматизированная установка с дистрибутива с использованием таких как методов как kickstart, принятый в дистрибутивах фирмы Red Hat или autoyast, принятый в дистрибутивах фирмы Novell?
- Какова будет технология обновления клиентского ПО?
- Как будут поддерживаться персональные настройки: централизованно или на каждой машине отдельно?

7.8.1 Метод развертывания новых клиентов

Существуют различные методы развертывания большого числа новых клиентских систем. Метод автоматической установки должен быть выбран до того как будет определена окончательная конфигурация клиентского ПО.

> ПОДСКАЗКА
>
> Определите метод развертывания до того как будет определена окончательная конфигурация клиентского ПО, поскольку метод развертывания может наложить ограничения на возможности конфигурирования.

Можно принять один из двух методов автоматизированного развертывания большого числа новых клиентов:

- Развертывание, основанное на применении готового образа диска или раздела.
- Использование включенной в дистрибутив процедуры автоматизированной установки ОС.

7.8.1.1 Метод установки на базе готового образа

Образ диска или устройства для установки можно создать с помощью такого инструмента как partimag (версия с открытым кодом популярного инструмента Symantec Ghost), информацию о нем можно найти по адресу:

http://sourceforge.net/projects/partimage/

Если принят метод установки с готового образа, то не приходится заботиться о включении в RPM-пакеты всех специфических для данной конфигурации настроек. С другой стороны, персональные настройки и уникальные свойства той или иной рабочей станции должны быть установлены с помощью специальных скриптов после установки ОС.

7.8.1.2 Метод автоматической установки с дистрибутива

Некоторые дистрибутивы включают метод автоматизированной установки ОС, из такой процедуры исключены все интерактивные моменты. В дистрибутивы фирмы Red Hat включена процедура «быстрого старта», kickstart. Если используется эта процедура, то поведение программы установки ОС, anaconda, контролируется не пользователем, а данными, которые заранее были сохранены в специальном файле. В дистрибутивах фирмы Novell есть приложение autoyast с аналогичными функциями. Оба эти метода установки основаны на применении пакетов в формате RPM, дополнительные, «домашние» разработки могут быть включены в них именно в этом формате, могут быть добавлены также и пост-инсталляционные скрипты.

Если используется метод автоматической установки с дистрибутива (основанный на применении RPM-пакетов), все дополнительные приложения и конфигурационная информация должны быть организованы в форме пакетов. Во-первых, надо убедиться, что это возможно, во-вторых, учесть, что формирование пакетов потребует дополнительного времени.

7.8.2 Технология обновления клиентского ПО

В зависимости от того, какой метод установки был принят, может различаться и метод обновления ПО на клиентской машине. Если использовался метод автоматической установки с дистрибутива, то все программное обеспечение было установлено из RPM-пакетов и для его обновления следует применять стандартные процедуры, принятые в этой технологии. Если же установка выполнялась при помощи готового образа диска, то система может включать приложения, не входящие в RPM-пакеты. Такие приложения требуют специфических процедур обновления.

Промышленные дистрибутивы обычно включают специальные инструменты для поддержки обновления пользовательского ПО.

7.8.3 Персональные настройки новых клиентов

После установки ОС на новых рабочих станциях необходима общесистемная настройка и персонализация. Это может быть сделано с помощью пост-инсталляционных скриптов и включать в себя:

- Имя машины
- IP адрес
- Удаленные файловые системы, монтируемые на этой рабочей станции
- Принтеры
- Специфические приложения

Персонализация включает в себя личные данные пользователя, который будет работать на конкретной машине. Ввод этих данных может осуществляться либо с помощью специального механизма, такого как программа firstboot, либо с помощью скриптов или приложений, которые пользователи сами запускают при первом входе в систему.

Механизм firstboot представляет собой комбинацию, во-первых, программы, которая стартует при первой загрузке системы и позволяет ввести учетную запись пользователя, задать дату и время, иногда дополнительные данные, и, во-вторых, специального файла, который создается после окончания работы программы и препятствует ее повторному запуску. Удобство этого механизма состоит в том, что для его использования не требуется входа в систему и учетная запись пользователя создается только после окончания его работы.

В том случае, если применение механизма firstboot невозможно, либо если пользователь уже имеет учетную запись, созданную централизовано, дополнительные персональные данные могут быть введены при помощи специальных скриптов и приложений. Для ввода некоторых идентификационных параметров могут потребоваться полномочия суперпользователя.

Существует ряд инструментов, которые позволяют автоматизировать перенос персональных настроек, которые были сделаны на клиентской машине под Windows, в новое окружение, под Linux. В этом разделе мы не будем говорить о них подробно, упомянем только один хорошо известный инструмент — пакет Versora Progression Desktop, информацию о котором можно найти в издании *Linux Client Migration Cookbook, Practical Planning and Implementation Guide for Migrating to Desktop Linux*:

http://www.redbooks.ibm.com/redpieces/abstracts/sg246380.html

7.9 Выявление неисправностей и техническая поддержка после миграции

После того как группа клиентов перейдет к работе под Linux, необходимо будет организовать для них техническую поддержку. Техническая поддержка должна включать не только помощь в решении обычных, повседневных задач, но и в преодолении постмиграционных проблем.

Проблемы могут быть не только технического плана. Привычные действия Windows-пользователя отличаются от тех, которые являются обычными под Linux.

При администрировании Linux-клиента используются методы, отличные от методов управления Windows-клиентом. То есть технический персонал также должен приспособиться к новым методам выявления и устранения неисправностей Linux-клиента. Первый и нелегкий шаг на этом пути — не перезагружать компьютер при возникновении малейшей проблемы, как это обычно делается с Windows-клиентом.

7.9.1 Каких сложностей следует ожидать

Главным средством предотвращения постмиграционных проблем является соответствующая подготовка конечных пользователей. Следует проявить внимание к пользователям и осторожность при их переходе к работе на Linux-клиентах.

Главные изменения для конечных пользователей при миграции на Linux-клиент будут следующие:

- Конечные пользователи не имеют больше полного контроля над своей файловой системой.
- Несмотря на то, что внешне рабочий стол Linux-клиента выглядит почти так же, как рабочий стол Windows, совсем другие процедуры применяются для изменения настроек рабочего стола и окружения.
- Меню, которые, как правило, появлялись при нажатии на правую клавишу мыши, сильно изменились по составу, а иногда их и вовсе нет.

- Изменились внешний вид и функциональность привычных приложений, а в некоторых случаях эти приложения просто заменены на другие.

Большинство этих изменений пройдут гладко, если организовать соответствующее обучение пользователей и снабдить их справочными материалами. Имеет смысл подготовить в удобной форме схемы, на котрых будут отражены различия в действиях пользователей при использовании старого клиента и нового. Такие схемы могут быть различны для рабочих мест с различным набором функций. Например, для рабочего места делопроизводителя, существенным моментом в работе которого является обмен данными по сети, изменится очень мало. А для работников, использующих продвинутый офис, изменения будут существенными.

7.9.2 Как справляться с неожиданными сложностями

Даже при условии, что конечные пользователи подготовлены к переходу на работу с новым клиентом, у них могут возникнуть определенные проблемы. Технический персонал и администраторы сети должны быть подготовлены к решению таких проблем.

Для того чтобы эффективно выполнять свои функции, технический персонал должен использовать в свой работе другие подходы, так как методы, которые эффективны для администрирования Linux сильно отличаются от методов, принятых в мире Windows.

Вообще говоря, решение проблем, возникающих в Windows, начинается со следующих действий:

- Перезагрузить систему
- Просмотреть протоколы
- Проверить драйверы оборудования и установить наиболее свежую версию из доступных

В отличие от монолита Windows, Linux состоит из большого числа во многом независимых модулей, следствием этого является очень большое число лог-файлов, в которых надо ориентироваться. Поэтому методика разрешения возникающих проблем будет следующая:

- Определить, работа какого модуля ответственна за возникшую проблему
- Просмотреть те протоколы, которые относятся к работе этого модули, и системный протокол
- Проверить, не содержат ли конфигурационные файлы ошибок или некорректных значений
- Перезапустить некорректно работающий сервис
- Проверить протокол после перезапуска сервиса

Технический персонал, работавший с Windows-клиентами, привык к тому, что половина проблем разрешается просто перезагрузкой системы. При работе с Linux-клиентом это не так. Как правило, просто перезагрузка системы ничего не дает, необходимо разобраться, в чем причина возникшей проблемы, и устранить ее.

Способ подготовки технического персонала к работе с новыми клиентами во многом зависит от уровня подготовленности и уровня задач, которые должен решать работник. Если в обязанности работника будут входить консультации пользователей и решение самых простых проблем, ему достаточно будет самому поработать с Linux-клиентом, а для тех, кто будет заниматься вопросами более сложного администрирования, необходимо специальное обучение.

7.9.3 Когда обращаться в службу поддержки производителя ПО

В службу поддержки поставщика дистрибутива или программного обеспечения следует обращаться, прежде всего, тогда, когда вы убедились, что проблема возникает из-за ошибки в работе ПО, или в том случае, когда программа работает не так, как должна работать согласно спецификации или руководству пользователя. Если проблема не связана напрямую с той частью операционной системы, за которую отвечает производитель дистрибутива или приложения, то решение можно попытаться найти, обратившись к ресурсам сообщества OSS.

Если у вас есть контракт на техническую поддержку с поставщиком дистрибутива, то правильно будет сначала обратиться за помощью именно туда, поскольку применение заплаток, которые можно найти в свободном доступе среди ресурсов OSS, к дистрибутиву, поставленному фирмой-производителем, может вызвать дополнительны проблемы. Попытаться же найти решение, обратившись за поддержкой к сообществу программистов, можно всегда.

Глава 8

Архитектура Linux и технические особенности

В этом разделе мы подробно рассмотрим набор программного обеспечения, входящего в операционную систему Linux, от нижних уровней до самых верхних, остановимся на том, что отличает Linux от других ОС, поговорим о свободном ПО и его отличиях от ПО с открытым кодом, дадим определение дистрибутива и обсудим важность стандартизации. Раздел состоит из двух частей:

- Разд. 8.1 Что такое Linux?

  Рассматриваются различия между ПО с открытым кодом и свободным ПО, обсуждаются дистрибутивы и стандарты.

- Разд. 8.2 Технические отличия

  Обсуждаются компоненты операционной системы Linux на всех уровнях, особое внимание уделяется отличиям Linux от других операционных систем.

8.1 Что такое Linux?

Автором операционной системы Linux является Линус Торвальдс. Сразу следует оговорить две вещи. Во-первых, Торвальдс является автором ядра Linux, во-вторых, в разработке с самого начала принимала участие группа программистов. В 1991 году Торвальдс разработал UNIX-подобное ядро с открытым кодом, присвоил версии номер 0.02 и поместил код ядра в сети. Вокруг этого проекта возникла группа, совместные разработки программистов продолжались, и в 1994 году на свет появилось ядро версии 1.0. Этот момент можно считать началом существования операционной системы Linux.

Роль ядра операционной системы состоит в том, чтобы обеспечивать работу основных аппаратных средств компьютера и взаимодействие с периферией. Ядро также управляет распределением оперативной памяти и запуском процессов. Масса дополнительного программного обеспечения необходима, чтобы реализовать все функции, которые обычно возлагаются на операционную систему. Большое число разработчиков ПО с открытым кодом сделали свой вклад в создание операционной системы Linux.

Сегодня Linux представляет собой полнофункциональную, гибкую и надежную, свободную операционную систему. Она работает на широком спектре аппаратных средств в различных вариантах архитектуры. Отличительной особенностью ОС Linux является большая свобода и гибкость в выборе ПО и настроек системы.

Как ядро Linux, так и значительная часть ассоциированного с ядром программного обеспечения, распространяется под лицензией GPL (GNU General Public License), таким образом исходные коды Linux доступны каждому. Полный текст и информацию о GPL можно найти в сети по адресу:

http://www.linux.org/info/gnu.html

Информация о Free Software Foundation и проекте GNU можно найти в сети по адресу:

http://www.fsf.org

8.1.1 Свободное программное обеспечение и программное обеспечение с открытым кодом

Критическое значение для создания и развития ОС Linux имели два течения в области разработки программного обеспечения:

- движение за свободное распространение программного обеспечение (точное определение свободного ПО можно найти по адресу: http://gnu.open-mirror.com/philosophy/free-sw.html)
- движение разработчиков программного обеспечения с открытым исходным кодом (точное определение с ПО с открытым кодом можно найти по адресу: http://en.wikipedia.org/wiki/Open_source)

С точки зрения стороннего наблюдателя разница между двумя этими движениями кажется несущественной, однако каждому, кто занят разработками программного обеспечения, ясно, что это совершенно разные вещи. Движение за свободное распространение ПО — это социальное движение, декларацией которого является свободный, бесплатный доступ и использование ПО, особенно инструментального. Но всякому, кто знаком с предметом, ясно, что программа не может быть по-настоящему свободной, если ее исходные коды недоступны Можно сказать, что понятие «ПО с открытым кодом» шире, чем понятие «свободное ПО». С другой стороны, если программа относится к ПО с открытым кодом, то это вовсе не значит, что она является свободной. Распространение ее может быть ограничено терминами той или иной лицензии и, вообще говоря, она может распространяться за плату. Движение разработчиков ПО с открытым исходным кодом — это техническое движение, декларацией которого являются определенные принципы в разработке ПО.

Тому, кто интересуется различиям между свободным ПО и ПО с открытым исходным кодом, мы советовали бы прочесть очень интересное эссе:

Live and let License, Joe Barr, linuxworld.com, published 5/23/2001

http://www.itworld.com/AppDev/350/LWD010523vcontrol4/

В рамках этих двух движений возникло большое число групп или сообществ разработчиков ПО, именно усилиями этих сообществ и достигнута та прозрачность и открытость, которые являются главными достоинствами Linux и которые можно видеть на всех его уровнях: от системного ядра до прикладных программ. Эти достоинства — не просто техническое отличие, они являются следствием той свободы, которая была заложена в основание Linux.

Именно в существовании движения за свободное распространение ПО и разработчиков ПО с открытым исходным кодом и коренится успех сегодняшних промышленных реализаций Linux для настольных вычислительных машин. Они же являются гарантией его жизнеспособности и динамического развития. Об этом тоже стоит подумать, планируя миграцию. Всякая организация, делающая стратегический вклад в Linux, делает вклад и в развитие общего прогрессивного движения, связанного с разработками ПО.

8.1.2 Дистрибутивы

Существуют сотни дистрибутивов Linux (вместо «Linux-дистрибутив» иногда используется сленговое «Linux-distro»), одни используются в конкретном регионе, другие распространены по всему миру. Дистрибутивом называют комплект, состоящий из ядра ОС, а также необходимых утилит и приложений вместе с процедурой установки этого комплекта. Некоторые дистрибутивы имеют специфический набор программного обеспечения, а некоторые предназначены для определенных видов аппаратных средств.

Среди наиболее известных дистрибутив можно назвать Red Hat, Novell SUSE, Debian, Gentoo и Mandriva. Как правило, организации, которые занимаются распространением дистрибутивов, обеспечивают также и различные формы поддержки пользователей.

Большинство современных дистрибутивов Linux предоставляют следующие возможности:

- Работа в 32-х и 64-х битовых архитектурах
- Многозадачность
- Поддержка многопользовательского режима и групп пользователей
- Необходимые средства для работы в сети
- Защита памяти
- Поддержка кластеров, включая выравнивание нагрузки и обработку сбоев

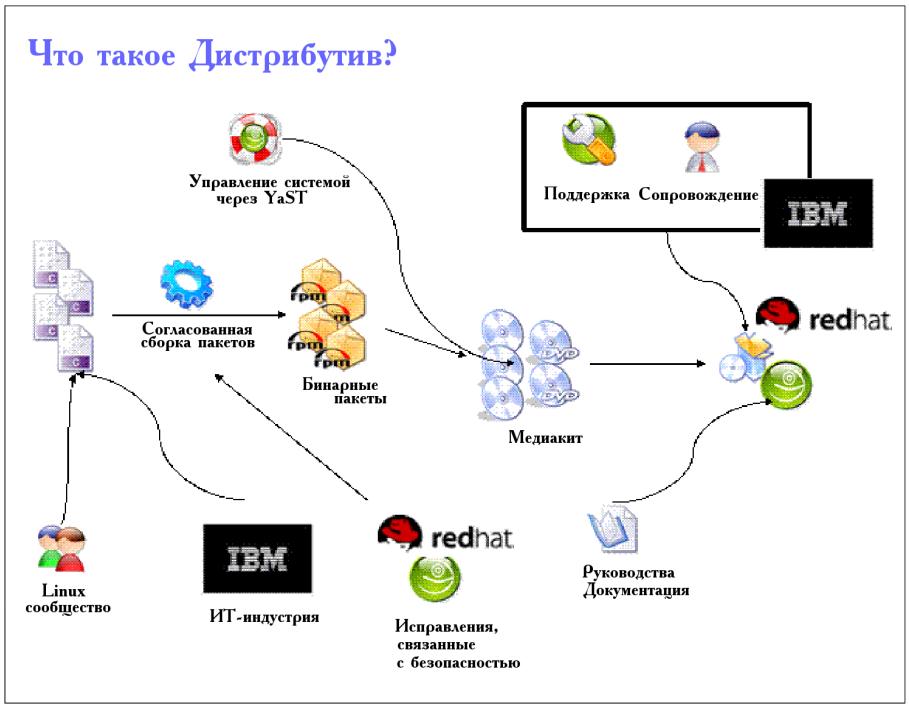

Рис. 8.1: Компоненты дистрибутива

- Динамическая перенастройка
- Поддержка архитектуры SMP

8.1.3 Стандарты

В последние годы масса усилий вкладывается в проекты стандартизации ядра Linux и его подсистем высокого уровня. Такие организации как FSG(Free Standards Group), FreeDesktop.org и OSDL (Open Source Development Labs) начали работать над многими проектами в этой области. В рамках этих проектов используются и расширяются уже существующие стандарты, такие как POSIX, Single UNIX Specification, XML, DOM, CORBA и многие другие. Вот некоторые примеры таких проектов:

- File system Hierarchy Standard (FHS)
- Linux Standard Base (http://www.linuxbase.org)
- Проект интернационализации (OpenI18N)
- Проект стандартизации печати (OpenPrinting)
- Специальные возможности
- Работа с кластерами (Open Cluster Framework)
- Data Center Linux
- Carrier Grade Linux

ЗАМЕЧАНИЕ

Не так давно был разработан и опубликован стандарт LSB версии 3.1. Подробности о LSB можно найти в Разд. 4.7.

Проекты стандартизации сфокусированы на функциональности низкого уровня, структуре файловой системы, новой системе печати, на стабилизации и разработке API для работы с кластерами и других системах. Ряд проектов направлен на решение проблемы бинарной совместимости программного обеспечения независимых производителей с различными дистрибутивами Linux.

С точки зрения стандартизации Linux для настольных машин прогрессирует достаточно быстро. Вот примеры наиболее существенных областей, где ведется разработка:

- Новые расширения и разбиение на модули X-сервера
- Cairo (система векторной графики) с возможностью вывода в X Window, OpenGL и Postscript/PDF
- HAL (прослойка для прозрачной поддержки различных аппаратных средств)
- GStreamer (потоковый мультимедийный пакет)
- D-BUS (канальная система обмена сообщениями)

8.2 Технические отличия

В этом разделе будут обсуждаться некоторые технические отличия в структуре и функциях тех версий ОС Windows и ОС Linux, которые ориентированны на настольные компьютеры.

Linux-клиент в общем является более динамично развивающейся системой, он имеет модульную структуру, его легко настраивать. ОС Linux состоит из ядра и окружающих ядро компонент. Некоторые возможности, которые есть также и в Windows, под Linux реализованы иначе. Тем читателям, которые привыкли работать в Windows-окружении, некоторые термины этого раздела могут быть незнакомы. Обратитесь к Прил. А.

Рассмотрим, из каких компонент состоит ОС Linux (см. также Рис. 8.2).

Ядро Linux
: Содержит критический набор сервисов, позволяющий запускать другие приложения и управлять аппаратным обеспечением. Драйверы устройств и модули могут динамически загружаться при необходимости и выгружаться. Ядро также отвечает за поддержку многозадачности и многопользовательского окружения

Устройства
: Устройство — это элемент внутри операционной системы, который обеспечивает взаимодействие с аппаратными средствами. Драйвер устройства — это программа, которая трансформирует обращения ОС в физические действия аппаратного обеспечения.

Модули
: Модулями ядра называют динамически загружаемые и выгружаемые части ОС, драйверы устройств часто являются модулями. Модули обеспечивают доступ ядра и приложений в функциональности аппаратного обеспечения. Загруженный модуль рассматривается как часть ядра, модуль выгружается, когда он перестает быть нужен, тем самым освобождая оперативную память.

Уровень приложений (пользовательское пространство)
: На этом уровне работают приложения, которые непосредственно взаимодействуют с пользователями. Взаимодействие может происходить в графическом окружении или посредством командной строки. Приложения получают доступ к сервисам ядра при помощи системных вызовов.

8.2.1 Процесс загрузки ядра

Главным отличием ОС Linux от ОС Windows является надежность. Linux обычно работает стабильно и не требует перезагрузок. В этом разделе мы рассмотрим в подробностях процедуру начальной загрузка ядра ОС Linux.

Процесс начальной загрузки ядра включает следующие этапы:

1. Инициализация процессора, доступны нижние адреса памяти, BIOS стартует загрузчик ОС (например, GRUB, LILO)
2. Загрузчик загружает `initrd` и ядро операционной системы в оперативную память, `bzImage` самораспаковывается в `/boot`

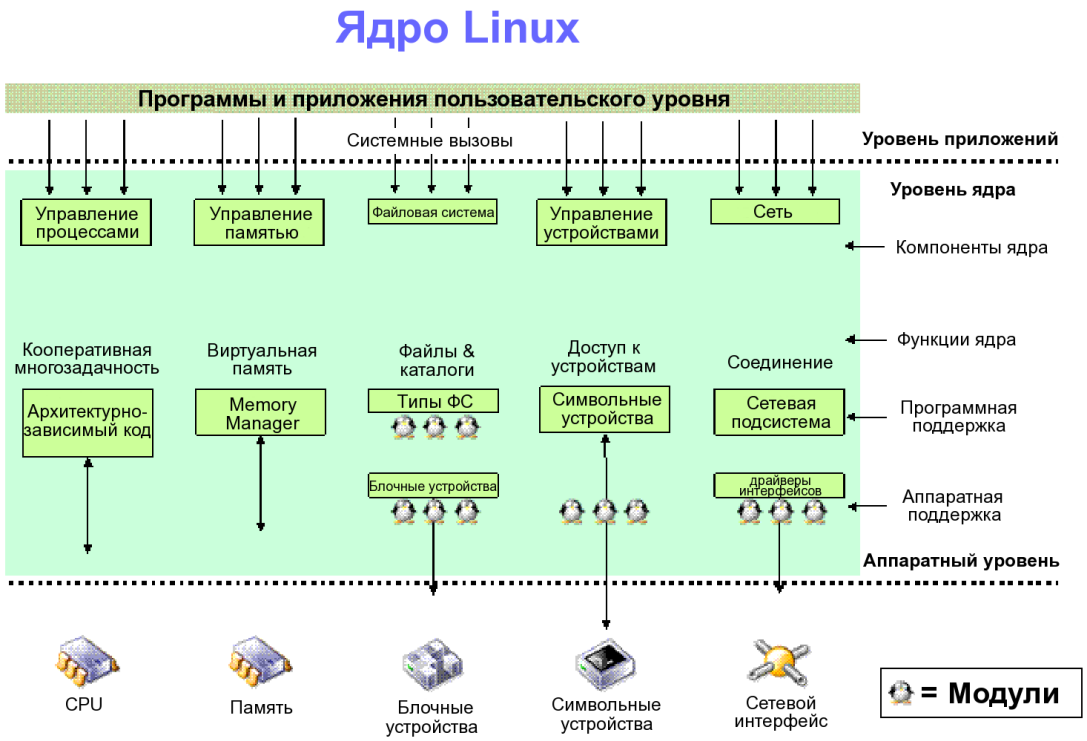

Рис. 8.2: Операционная система Linux

3. Загрузчик формирует параметры для запуска ядра (например, место расположения `initrd`) и передает управление ядру
4. Ядро распаковывает `initrd` и монтирует его как корневую файловую систему
5. Внутри `initrd` запускается исполняемый файл `linuxrc`. Файл `linuxrc` может быть любым допустимым исполняемым файлом, и initrd должен обеспечить выполнение этого файла (например, если `linuxrc` является скриптом оболочки, то в каталоге `/bin` должен быть соответствующий процессор обработки скрипта)
6. Программа `linuxrc` загружает модули, необходимые для монтирования реальной корневой файловой системы
7. Монтируется реальная корневая файловая система, `initrd` перемонтируется в `/initrd`
8. Стартует процесс init, ему присваивается PID 1, init — это единственный процесс, запускаемый непосредственно ядром, остальные процессы являются процессами-потомками по отношению к init

Следует сделать некоторые пояснения по поводу термина `initrd`. Это файл, который в сжатом виде содержит образ загрузочного RAM-диска, отсюда и происходит его название (initial RAM disk). После распаковки этот файл используется как временная корневая файловая система, которая дает возможность ядру системы выполнить стартовые операции. Временная корневая файловая система должна содержать все модули и драйверы, необходимые для монтирования реальной корневой файловой системы, например, если реальная корневая файловая система имеет тип ext3 и расположена на SCSI-диске, то initrd должен содержать модули для поддержки ext3 и SCSI.

8.2.2 Внешние взаимодействия, файлы и сервисы

Внешние взаимодействия в Linux осуществляются посредством сокетов. Сокет — это псевдо-файл, в который можно писать или из которого можно читать. Это соответствует операции получения или отправки данных для коммуникационных процессов. Например, коммуникация по сети между двумя Linux-системами построена на основе двух пар сокетов с каждой стороны, каждый сокет при этом связан с IP-адресом и портом, через который осуществляется взаимодействие.

Все внешние устройства в Linux также представлены для ОС в виде файлов, обычно эти файлы расположены в каталоге `/dev`. Запись данных в этот файл соответствует записи на аппаратное устройство. За трансформацию данных, записанных в файл устройства, в физическое действие аппаратного обеспечения отвечает драйвер устройства.

Как упоминалось выше в Разд. 8.1.3, файловая система Linux построена в соответствии с FHS (Filesystem Hierarchy Standard). Большинство дистрибутивов Linux придерживаются стандартной структуры основных каталогов, ниже можно видеть общепринятый вариант такой структуры

Таблица 8.1: Стандартная структура каталогов (FHS)

Каталог	Описание	Замечания
`/`	Корневой каталог файловой системы	Все остальные каталоги и файловые системы являются подкаталогами корневого каталога
`/bin`	Исполняемые файлы основных команд	Такие команды как sh, ls и т.д.
`/boot`	Статические файлы загрузчика ОС	Иногда представляет собой отдельную файловую систему
`/dev`	Файлы устройств	
`/etc`	Системные конфигурационные файлы	Состав каталога зависит от дистрибутива
`/lib`	Общие библиотеки и модули ядра	
`/media`	Точка монтирования для временно подключаемых медиа-устройств	Часто бывает подкаталогом `/mnt`
`/mnt`	Точка монтирования для временно подключаемых файловых систем	
`/opt`	Программы приложений	
`/sbin`	Исполняемые файлы системных команд	
`/srv`	Данные для сервисов	
`/tmp`	Временные файлы	
`/usr`	Головной каталог для вторичной иерархии	Содержит дерево подкаталогов, структура которого также регулируется FHS
/var	Основные переменные файлы	Файлы протоколов, PID, кэш-файлы и т.д.

Более подробное описание FHS можно найти в сети по адресу:

http://www.pathname.com/fhs

Существенным отличием от ОС Windows является отсутствие наименования жестких дисков буквами. Это отличие надо принимать во внимание, поскольку конечные пользователи привыкли, что символу `C:\`, например, соответствует системный диск ОС Windows, на диске `D:\` расположены пользовательские данные, а `E:` означает CD-ROM.

8.2.3 Многопользовательский режим

Работа с пользователями и группами в Linux построена во многом так же, как в UNIX. Это значит, что многопользовательский режим работы был заложен в ОС Linux с самого начала ее проектирования. В этом Linux существенно отличается от ОС Windows, которая была изначально рассчитана на работу одного пользователя и стала многопользовательской только пройдя долгий путь развития. Поскольку поддержка многопользовательского режима стала позднейшей надстройкой в ОС Windows для настольных компьютеров, то там она реализована далеко не так надежно, как в ОС Linux.

Чтобы уточнить разницу в подходах рассмотрим подробнее следующие вопросы:

- Могут ли несколько пользователей запускать процессы одновременно?
- Могут ли несколько пользователей открыть свою сессию на компьютере одновременно?
- Может ли быть установлено приложение, допускающее одновременное использование несколькими пользователями?

- Как устанавливаются полномочия для различных пользователей?

8.2.3.1 Могут ли несколько пользователей запускать процессы одновременно?

Систему можно считать по-настоящему многопользовательской в том случае, если она позволяет нескольким пользователям запускать процессы одновременно. В Linux такая возможность заложена в основание операционной системы, поскольку ряд системных процессов запускается только от имени суперпользователя. Формально, число одновременно работающих пользователей под Linux не ограничено, их число может быть лимитировано настройками системы.

Под Windows несколько пользователей могут запускать процессы одновременно в рамках Windows Terminal Server.

8.2.3.2 Могут ли несколько пользователей открыть свою сессию на компьютере одновременно?

Для обеих операционных систем ответом на этот вопрос будет «да», несколько пользователей могут войти в систему одновременно.

Если речь идет об использовании консоли с графическим дисплеем, то для Windows это возможно при использовании Windows Terminal Server и Remote Desktop Protocol (RDP). Под Linux возможно открыть сессию клиента для X-сервера на удаленном компьютере. Для рабочей станции под Linux есть также возможность переключения между консолями, которые открыты локально, с одним комплектом KVM (keyboard-video-mouse).

8.2.3.3 Может ли быть установлено приложение, допускающее одновременное использование несколькими пользователями?

И в этом случае для обеих операционных систем ответом на вопрос будет «да». Разница состоит, главным образом в том, как поддерживаются персональные данные и настройки, связанные с приложением.

В ОС Linux приложения обычно создают файлы настроек в домашнем каталоге пользователя, эти файлы имеют точку в качестве первого символа имени и поэтому называются «dot-files». Иногда место для хранения данных задается переменными окружения, заданными для каждого пользователя.

Изначально настройки для приложений в Windows хранились в системном регистре, который был один на всю систему, так как система была однопользовательской. Большинство современных версий ОС Windows поддерживают пользовательские настройки другим способом, файлы хранятся, главным образом, в папке «Documents and Settings», но старый подход до сих пор создает известные проблемы.

8.2.3.4 Как устанавливаются полномочия для различных пользователей?

Сразу оговоримся, что в момент написания этой книги нам ничего не было известно о том, какая система разграничения полномочий будет принята в версии Vista ОС Windows.

В ОС Linux существует один выделенный пользователь, он называется «суперпользователь» или «root». Многие системные процессы низкого уровня запускаются только от имени суперпользователя. Примером такого процесса может служить процесс начальной загрузки системы, init (всегда с PID 1). Суперпользователя нельзя переименовать, поэтому в каждой системе может быть только один суперпользователь, хотя часть его полномочий может быть делегирована другому пользователю.

В Windows всеми правами по доступу и модификации в системе обладает пользователь, который называется Администратор. При этом может быть несколько пользователей с правами, эквивалентными правам Администратора. Пользователь с административными полномочиями может изменить имя Администратор на какое-то другое имя. Такая возможность иногда создает весьма тяжелые проблемы при администрировании Windows-клиента.

Можно сказать, что Linux — это система с одним управляющим, а Windows — система с «семью няньками».

8.2.4 Графическое окружение и текстовый режим

ОС Windows является по своей сути графической системой. Windows всегда загружается только в графической моде, даже в тех случаях, когда для взаимодействия с компьютером не нужен графический режим, например, для управления работой сервера. Не говоря уже о том, что существует множество рабочих мест, для которых достаточен алфавитно-цифровой интерфейс, и необходимость использования графического окружения необоснованно повышает стоимость рабочего места. Большинство служебных программ для Windows имеют графический пользовательский интерфейс, что существенно затрудняет автоматизацию управления Windows-клиентом.

Как мы уже отмечали раньше, Linux хорош именно тем, что является модульной операционной системой. Хотя в большинстве дистрибутивов загрузка графического окружения выбрана как основной вариант входа в систему, это несложно изменить. Можно совсем не запускать графическое окружение, а пользоваться интерфейсом командной строки, такую настройку обычно используют на серверах. Часто к серверу вообще не подключают монитор, поскольку входить на него удобнее удаленно. В этом случае отсутствие графического окружения приводит к значительной экономии ресурсов, особенно оперативной памяти и дискового пространства. Все системные сервисы Linux построены на работе с командной строкой, что удобно для автоматизированного управления Linux-клиентом. Многие из сервисов имеют также графическую оболочку, которая позволяет выполнить те же действия в графическом режиме, более удобном для конечного пользователя. При этом, графических оболочек для одного инструмента может быть несколько.

Структура приложений графического окружения Linux также модульная, то есть, существует возможность независимо стартовать, останавливать и заменять его компоненты. В отличие от классической ситуации в Windows Linux оставляет возможность выбора средств обеспечения графического интерфейса на всех уровнях. Ниже в Табл. 8.2 представлены варианты решений на различных уровнях пользовательского интерфейса.

Таблица 8.2: Уровни системы X Window

Уровень	Выбор
Менеджер дисплея	XDM, GDM, KDM, Entrance
X Window-сервер	XFree86, X.org
Диспетчеры окон	FVWM, IceWM, WindowMaker, Metacity, KWin, Enlightenment
Инструментарий GUI	OpenMotif, GTK+, Qt, wxWidgets, FLTK, FOX, Swing, SWT, EFL
Графическое окружение рабочего стола	KDE, GNOME, GNUStep, XFCE, ROX, Looking Glass, Metisse

В том случае, если выбран графический вариант входа в систему, менеджер дисплея запускается как один из сервисов во время начальной загрузки системы. Менеджер дисплея запускает сервер X Window и в дальнейшем взаимодействует с ним. Менеджер дисплея отвечает за появление графического приглашения к регистрации и предлагает пользователям выбрать для их сессии менеджера рабочего стола в том случае, если в системе возможен такой выбор.

ЗАМЕЧАНИЕ

Возможность в общем случае выбрать менеджера рабочего стола (например, KDE или GNOME) есть наглядная демонстрация модульности и гибкости Linux, которые отличают эту ОС от ОС Windows.

Система X Window по своей сути является системой типа клиент-сервер. Однако способ взаимодействия в ней сервера и клиента на первый взгляд кажется противоречащим интуиции. Та часть, которая запускается локально на рабочей станции, называется X Сервер, а запускаемое удаленно приложение, которое больше похоже на традиционный сервер, называется X-клиент. Ниже на Рис. 8.3 схематически показана архитектура системы X Window. X-сервер контролирует поведение дисплея и пары мышь-клавиатура. X-клиенты, в нашем примере это программы Xterm и Xclock, могут быть запущены либо на локальной машине (компьютер A),

либо на удаленной машине (компьютер B) и взаимодействуют с X-сервером посредством сокетов.

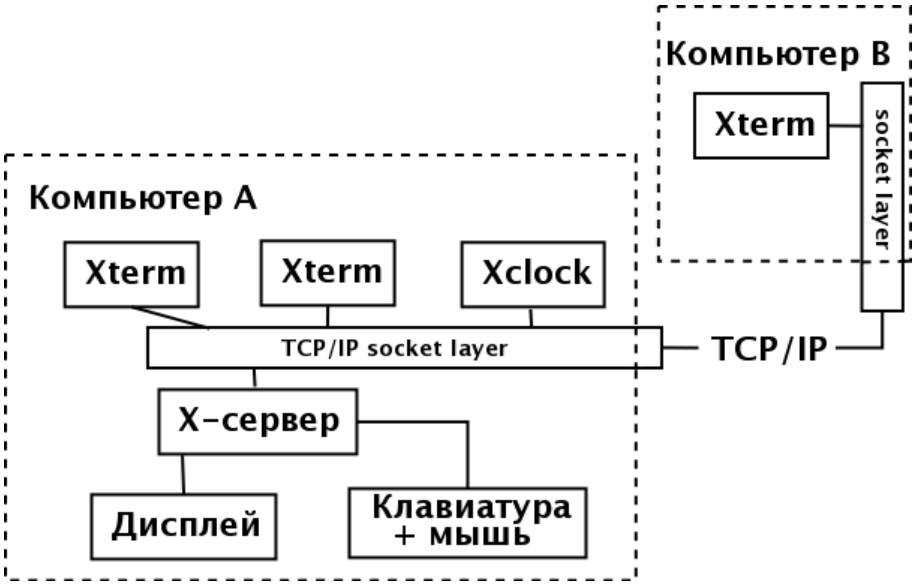

Рис. 8.3: Схема архитектуры системы X Window

X-сервер обеспечивает прорисовку изображений на экране, используя информацию о видеокарте и драйвер монитора, и контролирует сигналы, приходящие от устройств ввода данных, таких как клавиатура и мышь. Интерфейс между X-сервером и клиентом регулируется стандартом X11, поэтому любой X-сервер может принять сигналы от любого X-клиента. Интерфейс работает с использованием сетевых сокетов даже в том случае, если и клиент, и сервер находятся на одной локальной машине, скорость обмена в этом случае, конечно, выше, чем в случае обмена посредством локальной сети. Такой уровень абстракции позволяет X-клиенту взаимодействовать с X-сервером однотипно, вне зависимости от их взаимного расположения.

Диспетчер окон (http://xwinman.org) — это клиентская программа системы X Window, которая обеспечивает X-клиентам возможность манипулировать окнами: позиционировать, передвигать, изменять размер. Диспетчер также выводит заголовок окна, рамки, обеспечивает работу бегунков и других сопутствующих элементов, следит за фокусом и обрабатывает события, передаваемые от устройств ввода. Существует возможность протестировать работу диспетчера окон и оконного интерфейса. Для этого с помощью Xnest запускают вложенный X-сервер (который в такой ситуации является одновременно и X-клиентом):

Пример 8.2.1: Запуск вложенного X-сервера с помощью Xnest

```
Xnest -fp `xset -q | grep fonts` :1 &
xterm -display :1 &
```

На основе базовых библиотек X11 было разработано множество инструментов, облегчающих программирование графического пользовательского интерфейса GUI. Эти библиотеки позволяют разрабатывать темы рабочего стола и дополнительные библиотеки, обеспечивающие однотипный внешний вид и функциональность различных приложений. Кроме того, существуют и интегрированные среды разработки IDE (Integrated Development Environment), которые позволяют сделать процесс создания графического интерфейса еще более технологичным, например:

- KDevelop/QTDesigner для KDE/Qt
- Anjuta/Glade для GNOME/GTK+
- MonoDevelop для Mono/C#
- Eclipse или NetBeans для Java Swing/SWT

Менеджер рабочего стола обладает большими возможностями, чем диспетчер окон и предоставляет пользователю графическую среду офисного плана, в которой естественно запускаются такие приложения как:

- Электронная почта и средства делового планирования
- Непосредственный обмен сообщениями
- Работа с изображениями
- Создание, просмотр и редактирование документов
- Сканирование
- Чтение и запись CD/DVD.

Все эти приложения имеют однородный по функциям и внешнему виду интерфейс, широкие возможности настройки и взаимодействия, включая функциональность «drag and drop», управление сессиями и виртуальными рабочими столами.

Менеджеры рабочего стола KDE и GNOME (вместе с соответствующим инструментарием Qt и GTK+) постепенно становятся стандартными для большинства разработчиков графических приложений под Linux. Некоторые традиционные поставщики UNIX также начинают их использовать.

8.2.5 Системные уровни выполнения

ОС Linux имеет возможность загружаться с различными уровнями выполнения. Системный уровень выполнения является конфигурационной опцией, которая определяет, какие процессы будут выполняться при загрузке системы для каждого уровня. Некоторый аналог уровней выполнения Linux представляет собой загрузка в безопасном режиме под Windows. Разница состоит в том, что уровни запуска позволяют управлять системой более избирательно и что поведение системы на каждом уровне можно гибко настраивать. Существует возможность загрузиться с одним из 11 системных уровней, для их обозначения используются цифры от 0 до 9 и литера S, однако широко используются только уровни 1,2,3 и 5. Уровень 0 зарезервирован для остановки системы, а уровень 6 — для перезагрузки. Уровень выполнения S — это специальный уровень, он используется скриптами до того, как начат запуск процессов «обычного» уровня выполнения. Общеупотребительные уровни выполнения и соответствующие им режимы можно видеть в таблице ниже.

Таблица 8.3: Уровни выполнения и соответствующие режимы

Уровень выполнения	Режим
1	Однопользовательский режим
2	Многопользовательский текстовый режим без поддержки сетевых соединений
3	Многопользовательский текстовый режим с поддержкой сетевых соединений
5	Многопользовательский режим с поддержкой сетевых соединений и запуском графического окружения

Конфигурация основных уровней запуска записана в файле `/etc/inittab`, суперпользователь может переключить уровень выполнения системы командой **init**.

8.2.6 Диски, разделы и файловые системы

В отличие от ОС Windows в ОС Linux не принято считать каждый отдельный диск узлом верхнего уровня в иерархии файловой системы. В Linux существует единственная точка верхнего уровня, каталог с именем «/», все остальные устройства монтируются в его подкаталоги. Например, CD-ROM монтируется как `/mnt/cdrom` (или как `/media/cdrom` в некоторых дистрибутивах). Оба вида организации файловой структуры имеют свои достоинства и недостатки. В Windows сразу можно видеть, какой файл физически расположен на каком устройстве. В Linux структура скорее логическая, не ориентированная на конкретную конфигурацию аппаратных средств, например, каталог `/home` может быть на том же диске, что и системная точка монтирования (`/`), а может быть и на другом диске, для пользователя при этом не будет никакой разницы.

В Windows сетевые устройства монтируются также, как и диски. В Linux другие файловые системы, в том числе сетевые (например, NFS или SMB) монтируются в подкаталоги текущей файловой системы (см. Прим. 8.2.2).

Например, существует возможность разместить домашний каталог пользователя на сетевом устройстве, такой вариант будет напоминать использование роуминг-профиля в Windows.

Пример 8.2.2: Монтирование Samba-каталога

```
smbmount //itsont05/data /mnt/itsont05_data
```

Linux может работать с большим числом файловых систем, таких как ext2, ext3, ReiserFS, XFS и JFS. Каждая из них имеет свои достоинства, в некоторых случаях использование той или иной системы может вызвать сложности, но все они поддерживаются ОС. Основной файловой системой для дистрибутивов Red Hat является ext3, а в Novell Linux Desktop используется ReiserFS. Обе этих системы являются системами журналируемого типа, то есть поддерживают специальный журнал (log), в который записываются последние изменения, это позволяет без проблем восстановить файловую систему в случае того или иного сбоя. Файловые системы, с которыми работает Linux, имеют встроенные средства предупреждения фрагментации диска. Даже в том случае, если диск оказывается фрагментирован, это не оказывает большого влияние на скорость дисковых операций.

Linux также поддерживает и FAT, файловую систему, использующуюся в DOS и Windows. Файловая система NTFS, которая появилась вместе с Windows NT, поддерживается только в режиме чтения. Существуют драйверы, которые позволяют писать на NTFS, но они работают не очень надежно.

8.2.7 Виртуальная память

Обе операционные системы как Windows, так и Linux, имеют систему работы с виртуальной памятью, но способы поддержки виртуальной памяти у них различны. Впрочем, изменения бывают даже в различных версиях одной ОС.

Основное различие между Linux и Windows состоит в подходе к вопросу, когда начинать использование виртуальной памяти. Linux начинает перемещать данные в область свопинга только тогда, когда исчерпана вся оперативная память. Обратное перемещение возникает в том случае, если выгруженная страница требуется активному процессу. Большой объем свопинга снижает производительность системы, особенно тогда, когда перемещение происходит в обоих направлениях. Активный свопинг может быть следствием недостатка оперативной памяти, некорректно установленных параметров ядра, отвечающих за работу с виртуальной памятью, или, возможно, неправильно выбранным размером страницы для свопинга. В Windows принята система превентивного свопинга, то есть, использования виртуальной памяти до того, как исчерпана вся физическая оперативная память. Это ускоряет освобождение оперативной памяти тогда, когда она требуется активному процессу, но увеличивает частоту выгрузки и объем перемещаемых данных.

Другим существенным отличием является то, что в Windows используется своп-файл, а в Linux — своп-раздел, хотя и своп-файлы также могут использоваться. В Windows системный диск C: обычно содержит файл `pagefile.sys`, который используется для размещения данных, выгруженных из оперативной памяти. Его размер ограничен только размерами жесткого диска и вследствие этого файл может быть сильно фрагментирован, что замедляет работу операционной системы. В Linux во время установки системы для свопинга создается специальный раздел. Этот раздел не используется в основной работе ОС и зарезервирован только для размещения страниц основной памяти, поэтому он не может быть фрагментирован. С другой стороны, размер раздела определяется при первичной разметке диска и для того, чтобы увеличить область свопинга, требуется переразмечать диск.

ПОДСКАЗКА

Если на машине установлены обе операционные системы, то, с целью повышения эффективности использования дискового пространства, имеет смысл настроить конфигурацию виртуальной памяти в Windows и Linux так, чтобы использовался один и тот же раздел. Информацию о том, как это сделать, можно найти в сети по адресу:

http://www.tldp.org/HOWTO/Swap-Space.html

В Linux на одной и той же машине можно отвести несколько разделов под своп и создать несколько своп-файлов. Такая возможность может быть использована, например, в том случае, если созданный при первичной разметке раздел оказался слишком мал. В отличие от ОС Windows, где размер файла, предназначенного для хранения выгруженных страниц, увеличивается автоматически, размер раздела в Linux не может увеличиться сам собой, администратор системы должен позаботиться об этом в случае необходимости.

ПОДСКАЗКА

Размещение отдельных разделов или файлов для свопинга на различных физических дисках, установленных в одной и той же системе, заметно повышает производительность системы.

8.2.8 Ссылки

В отличие от ОС Windows ОС Linux поддерживает ссылки на файлы (или каталоги) на уровне файловой системы. Вообще говоря, файловая система NTFS также имеет механизм поддержки ссылок, но Microsoft Windows не использует его. Существуют два типа ссылок: жесткие ссылки и символьные ссылки.

Символьная ссылка (иногда их называют «симлинки» от symlinks) представляет собой просто специальный файл, единственным содержанием которого является строка, которая содержит указатель на реальный файл (или каталог), при этом символьная ссылка может указывать на объект, расположенный на другом разделе. Символьная ссылка напоминает краткие имена Windows. Сходство между ними еще и то, что при удалении файла символьные ссылки на него не удаляются, но они перестают работать.

Для того, чтобы понять, что такое жесткая ссылка в Linux, надо уяснить себе, что операционная система работает не с известными нам именами, а с идентификаторами фалов inode, в которых записана вся необходимая ОС информация о расположении, размере и свойствах файла. Имена фалов являются только указателями на inode, они введены потому, что человеку было бы сложно узнавать файл по системному идентификатору. Жесткая ссылка — это просто еще один указатель на тот же inode. Жесткие ссылки имеют ограничения: нельзя создать ссылку в каталоге, который сам находится в другой файловой системе, нельзя создать ссылку на каталог, во избежание возникновения циклов в файловой системе. Если существует несколько жестких ссылок, то есть указателей на inode, то удаление одной их них никак не повлияет на существование файла, но после удаления последней жесткой ссылки файл будет удален из файловой системы.

8.2.9 Замена ядра

В отличие от Windows, которая представляет собой монолит, Linux построен по модульному принципу, вообще говоря, ядро его тоже представляет собой некоторый модуль. Вы можете установить на своей машине несколько ядер (хотя только одно из них будет работать в каждый момент времени). Например, устанавливая систему с дистрибутива, вы установили то ядро, которое в нем содержалось, но это ядро не поддерживает новое аппаратное обеспечение или специфическое устройство. Вы нашли свежую версию ядра, в которой нужное вам устройство поддерживается. Для того, чтобы проверить работоспособность оборудования с новым ядром, можно установить его дополнительно к старому и использовать опцию выбора загружаемой системы в загрузчике (GRUB, lilo и т.п.).

Следует иметь в виду, что в стандартных конфигурациях некоторых дистрибутивов возможность выбора системы для загрузки отключена. Например, дистрибутив Novell/SuSE настроен так, загружается только одно ядро, то, которое было установлено в системе последним. Это естественный подход, поскольку большинству пользователей никогда не приходится заниматься заменой ядра своей ОС. Для того, чтобы иметь возможность загружать различные версии ОС, необходимо изменить настройки загрузчика GRUB. Дополнительную информацию о системных загрузчиках и о том, как организовать возможность загрузки нескольких ядер, можно найти в сети по адресам:

http://www.gnu.org/software/grub/

http://en.wikipedia.org/wiki/GNU_GRUB

http://www.tldp.org/HOWTO/Multiboot-with-GRUB.html

8.2.10 Драйверы устройств и поддержка аппаратного обеспечения

Как известно, сильной стороной ОС Windows является поддержка широкого спектра аппаратных устройств и периферии. Так как Windows была более распространена в качестве ОС для настольных компьютеров, то производители периферийного оборудования разрабатывали в первую очередь драйверы для Windows, до других ОС дело иногда и не доходило.

Сейчас ситуация меняется. Все больше производителей аппаратных средств предоставляют исходные коды своих драйверов, соответственно, устройства могут быть использованы в любой ОС, независимо от версии.

Однако надо иметь в виду, что некоторое оборудование может не иметь драйвера под Linux. Если такое оборудование распространено широко, то скорое появление драйвера под Linux весьма вероятно, поскольку сеть пользователей и разработчиков ПО для Linux все время расширяется, и драйверы возникают благодаря усилиям сообщества разработчиков ПО с открытым исходным кодом.

8.2.11 Поддержка шрифтов

Как уже упоминалось выше, ОС Windows может работать только в графическом режиме, и драйвер дисплея интегрирован в систему, то же можно сказать и о поддержке шрифтов. В Windows используется множество шрифтов, некоторые из них имеют копирайт фирмы Microsoft.

В ОС Linux дисплеем управляет система X Window. Текущая версия системы X Window имеет отдельный шрифтовой сервер. Шрифтовой сервер может использоваться не только X Window, но и другими приложениями. Для Linux существует широкий набор шрифтов, как свободно распространяемых, так и тех, которые можно получить за плату. В состав дистрибутивов Novell Linux Desktop и Red Hat Desktop входят некоторые лицензионные шрифты.

8.2.12 Поддержка 64-разрядных и многоядерных процессоров

До недавнего времени поддержка 64-разрядных процессоров редко включалась в ОС для настольных машин. Сейчас уже не только 64-разрядные, но и многоядерные процессоры все более широко используются в рабочих станциях.

Поддержка многоядерных процессоров и в Linux, и в Windows, организована также как и поддержка нескольких CPU, в итоге работа операционной системы происходит с набором логических CPU.

Надо отметить, что в Linux поддержка 64-разрядных процессоров появилась гораздо раньше, чем в Windows, например, он давно работал на 64-разрядном процессоре архитектуры IBM Power.

Переходя на 64-разрядную версию ОС, надо помнить, что не все 32-разрядные приложения гладко перенесут этот переход, поскольку они могут требовать для своей работы 32-разрядных системных библиотек. Использование драйверов устройств для 32-разрядной системы может также привести к возникновению проблем в 64-разрядной ОС.

Часть IV

Пилотная миграция

Глава 9

Модели развертывания клиентов

В этой главе будут подробно обсуждаться примеры организации рабочих мест под Linux, эти примеры могут оказаться полезны при планировании любой пилотной миграции. Мы будем обсуждать среду KDE Kiosk, пользовательскую настройку GNOME и создание полноценного самоуправляющегося Linux-клиента на базе Red Hat. Близкие к этим темам вопросы, связанные с автоматизацией, будут обсуждаться также в Прил. C.

Эта глава содержит следующие разделы:

- Разд. 9.1 Среда KDE Kiosk

  Обсуждаются возможности пакета KDE Kiosk framework для упрощения рабочего стола, сокращения числа доступных функций и защиты настроек от возможных изменений при использовании интегрированной среды KDE

- Разд. 9.2 Конфигурирование GNOME

  Обсуждаются способы конфигурирования рабочей среды GNOME и возможности утилиты gconftool-2

- Разд. 9.3 Инструменты удаленного доступа

  Обсуждаются методы установки полноценных самоуправляющихся Linux-клиентов на всем предприятии

9.1 Среда KDE Kiosk

В этом разделе будет продемонстрировано, как с помощью пакета KDE Kiosk framework, который включен в KDE3, можно ограничить и зафиксировать функциональность рабочего стола. Мы покажем, каким образом можно приписать те или иные профили пользователям или группам пользователей, пометить элементы конфигурации как неизменяемые, ограничить выбор функций и доступ к ресурсам сети. Это можно выполнить прямым редактированием конфигурационных файлов или используя удобный инструмент с графическим интерфейсом пользователя Kiosk Admin Tool, дополнительную информацию о котором можно найти в сети по адресу:

http://extragear.kde.org/apps/kiosktool

Подробности о том, как в KDE организовано хранение данных пользовательского профиля, можно найти также в Разд. B.3.

9.1.1 Пользовательские профили

Свежая версия KDE дает пользователю массу возможностей для изменения свойств рабочего окружения в соответствии со своими вкусами. Внутри предприятия существуют такие рабочие места, где это свойство становится нежелательным. Специально для этих случаев в KDE3 был включен пакет KDE Kiosk framework, который позволяет отключать лишние возможности KDE и создавать более контролируемое рабочее окружение. KDE Kiosk создает дополнительную надстройку над конфигурационными файлами KDE и добавляет простой API, к которому обращаются приложения, когда им нужно получить авторизацию для некоторых операций.

Этот окружение должно использоваться вместе со стандартными мерами безопасности, принятыми в Linux. Например, доступ на запись к некоторым конфигурационным файлам KDE Kiosk должен быть только у администратора, который сам может быть как суперпользователем, так и обычным пользователем, созданным специально для целей администрирования среды KDE Kiosk. Фактически используется стандартная семантика пользователей и групп, принятая в UNIX, KDE Kiosk добавляет над ней еще один конфигурационный уровень. Пользовательский профиль KDE Kiosk, задающий набор ограничений, может быть приписан отдельному пользователю или группе пользователей.

Вы должны сообщить KDE в основном конфигурационном файле `/etc/kderc` (другим вариантом может быть `/usr/share/config/kdeglobals`), какие определены профили, и где будет расположен файл, приписывающий профили определенным пользователям. Ниже приведен пример конфигурационного файла, в котором определены два различных пользовательских профиля, содержащие интернационализацию (Английскую и Немецкую[de]), и оба они принадлежат суперпользователю root (ProfileInstallUser=root). Остальные три профиля идентичны.

Пример 9.1.1: Профили и файлы соответствия, заданные в /etc/kderc

```
[Directories]
kioskAdmin=root:
profileDirsPrefix=/etc/kde-profile/
userProfileMapFile=/etc/kde-user-profile

[Directories-default]
ProfileDescription=Default profile
ProfileDescription[de]=Standard Profil
ProfileInstallUser=root
prefixes=/etc/kde-profile/default/

[Directories-ITSO]
ProfileDescription=ITSO Test Profile
ProfileDescription[de]=ITSO Standard Profil
ProfileInstallUser=root
prefixes=/etc/kde-profile/ITSO/
```

Для миграции внутри своей организации IBM Technical Support Organization (ITSO) с использованием KDE Kiosk мы определили пять различных профилей и приписали им имена default, Redbook, ITSO, Designer, Administrator и соответствующие каталоги для конфигураций `/etc/kde-profile/[имя профиля]`. Ниже в Табл. 9.1 можно видеть эти профили, которые мы создали для своего тестового варианта миграции, и приписанную каждому профилю роль.

Таблица 9.1: Профили и расшифровка ролей

Профиль	Роль
default	Обычный пользователь, никакой специфической роли
Redbook	Один из авторов Redbook
ITSO	Сотрудник ITSO
Designer	Сотрудник ITSO, выполняющий дизайнерские функции
Administrator	Имеет неограниченный доступ

Следующим шагом мы должны были приписать профили нашим пользователям и группам пользователей. Для этого можно использовать KDE Kiosk Admin Tool с графическим интерфейсом, как показано на Рис. 9.1, редактировать конфигурационный файл (см. Прим. 9.1.2). Дополнительную информацию о KDE Kiosk Admin Tool можно найти в сети по адресу

http://extragear.kde.org/apps/kiosktool

Пользователю в KDE Kiosk можно приписать более, чем одну роль, например, можно приписать

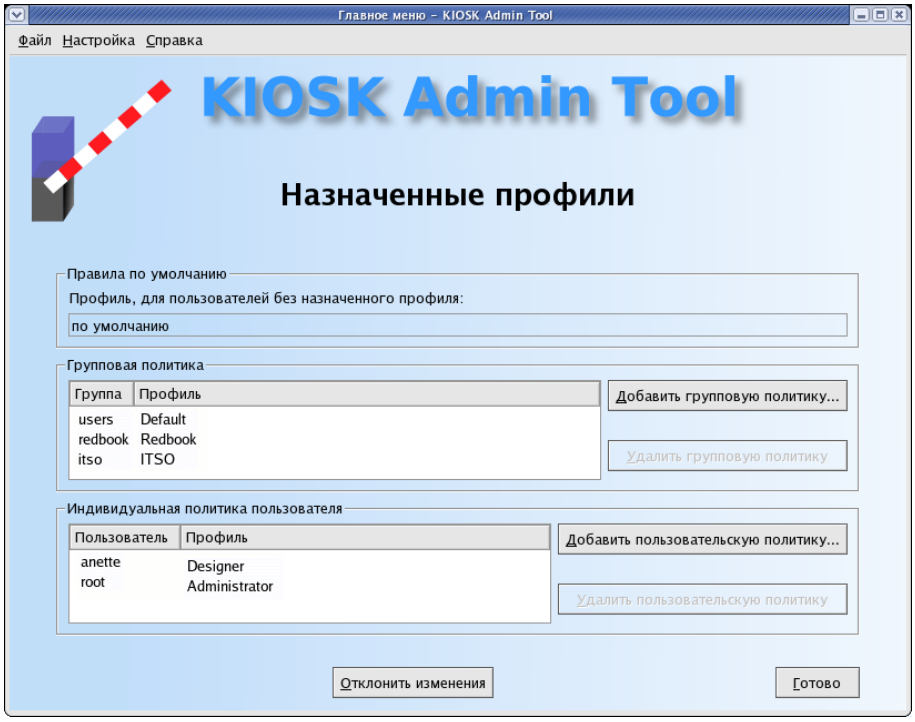

Рис. 9.1: Профили пользователей в окне KDE Kiosk Admin Tool

```
amelie=Administrator,Designer
```

При этом свойства того профиля, который указан первым, будут иметь приоритет в случае конфликтных ситуаций. То же самое правило работает и для групп. Например, можно приписать

```
partner=Partner,Redbook
```

для группы partner, при этом профиль Partner будет иметь больший приоритет.

Пример 9.1.2: Установка соответствия между профилями и пользователями в файле /etc/kde-user-profile

```
[General]
groups=itso,redbook,users

[Groups]
itso=ITSO
redbook=Redbook
users=default

[Users]
anette=Designer
root=Administrator
```

Другой интересный случай — когда пользователь, не включенный в раздел Пользователи (Users), входит в состав различных UNIX-групп, внесенных в различные профили в разделе Группы (Groups). В этом случае ранее зарегистрированные профили будут иметь более высокий приоритет. Последнее, но не менее важное: если пользователь включен в раздел Пользователи (Users), только профили этой записи принимаются во внимание (то есть пользователи, являющиеся членами UNIX-групп и включенные в профайл в этом случае роли не играют).

Несмотря на то что эти правила однозначно определяют права того или иного пользователя, в большинстве случаев достаточно простой и даже линейной иерархии профилей. Например, сначала можно создать основной профиль, который будет использоваться в большинстве случаев, затем добавить некоторые возможности для

группы, для которой требуются большие привелегии, и так далее. Если обнаруживается, что некоторая группа не удовлетворяет этой иерархии, создайте для нее профиль отдельно и добавьте его в самую первую запись или отредактируйте конфигурационный файл.

Хорошим свойством Kiosk Admin Tool является возможность его удаленной конфигурации. На Рис. 9.2 показано, как можно сопоставить локальные каталоги и удаленные (при этом вы имеете возможность изменить базовый URL). Файлы пересылаются при помощи протокола SSH с обычной семантикой типа `[fish://]` или каким-либо другим сетевым протоколом, поддерживаемым KDE.

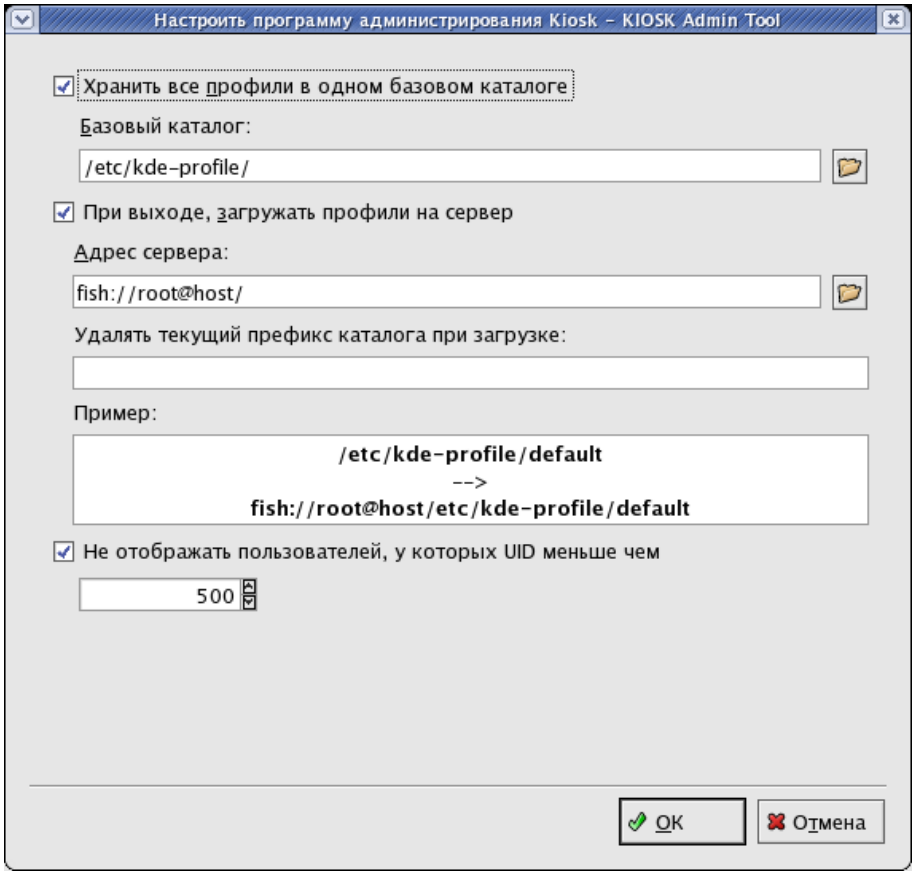

Рис. 9.2: Удаленное конфигурирование при помощи KDE Kiosk Admin Tool

9.1.2 Опции блокировки

Рассмотрим, какие опции доступны для настройки в среде KDE Kiosk.

Опции общей конфигурации:

- Отключить контекстное меню оконного менеджера (Disable Window Manager context menu) (Alt-F3)
- Отключить закладки (Disable Bookmarks)
- Отключить все задачи и приложения, которые требуют права суперпользователя (Disable all tasks and applications that require root access)
- Отключить доступ к командной строке (Disable access to a command shell)
- Отключить возможность завершения сеанса (Disable Logout option)
- Отключить возможность блокировки экрана (Disable Lock Screen option)
- Отключить пункт «Выполнить» (Disable «Run Command» option) (Alt-F2)
- Отключить перемещение панелей инструментов (Disable toolbar moving)
- Отключить запуск произвольных `.desktop` файлов (Disable execution of arbitrary `.desktop` files)

- Отключить запуск второго сеанса X (Disable starting of a second X session)
- Отключить историю строки ввода (Disable input line history)

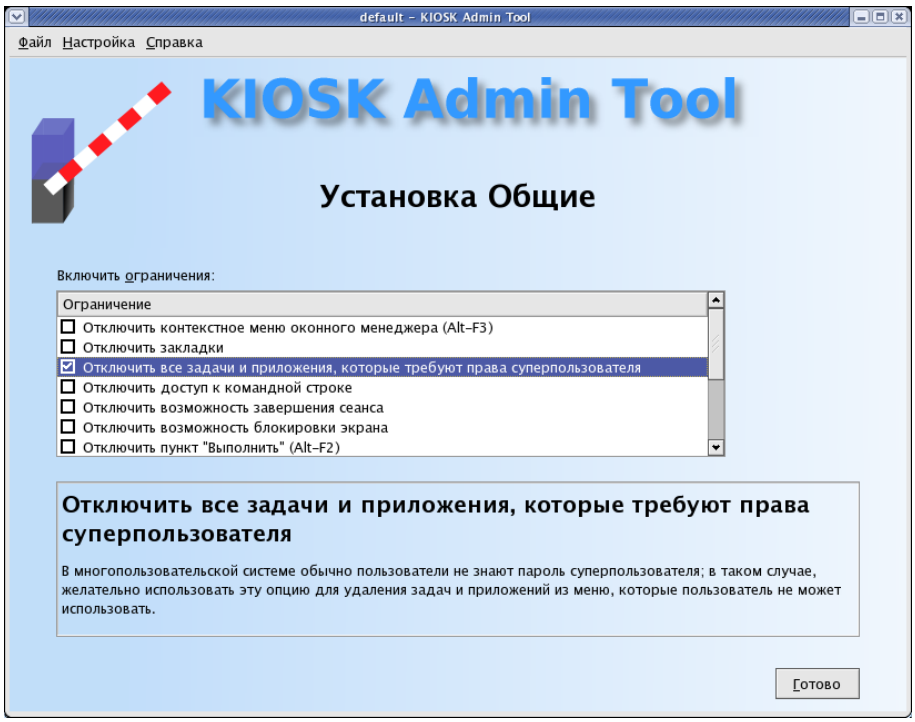

Рис. 9.3: Настройка опций общей конфигурации в Kiosk Admin Tool

При выборе настроек следующей секции можно включить режим предварительного просмотра, при этом добавлять апплеты, изменять фоновый рисунок или цвет фона, манипулировать пунктами меню и добавлять иконки на второй рабочий стол и сразу же видеть результаты.

- Опции конфигурации значков на рабочем столе
  - Заблокировать настойки Рабочего стола (Lock down Desktop Settings)
  - Отключить контекстные меню (Disable context menus)
  - Заблокировать все значки Рабочего стола (Lock down all Desktop icons)
- Опции конфигурации фона рабочего стола
  - Заблокировать фон рабочего стола (Lock down Desktop Background Settings)
- Опции конфигурации хранителя экрана
  - Заблокировать хранитель экрана (Lock down Screen Saver Settings)
  - Отключить хранители экрана, использующие OpenGL (Disable OpenGL-based Screen Savers)
  - Только безопасные хранители экрана (Discreet Screen Savers Only)
- Опции конфигурации меню KDE
  - Отключить все задачи и приложения, которые требуют права суперпользователя (Disable all tasks and applications that require root access)
  - Отключить возможность изменения меню (Disable menu editing)
- Настройка конфигурации темы рабочего стола
  - Заблокировать настройку стиля (Lock down Style Settings)
  - Заблокировать настройку цветов (Lock down Color Settings)
  - Заблокировать настройки шрифтов (Lock down Font Settings)

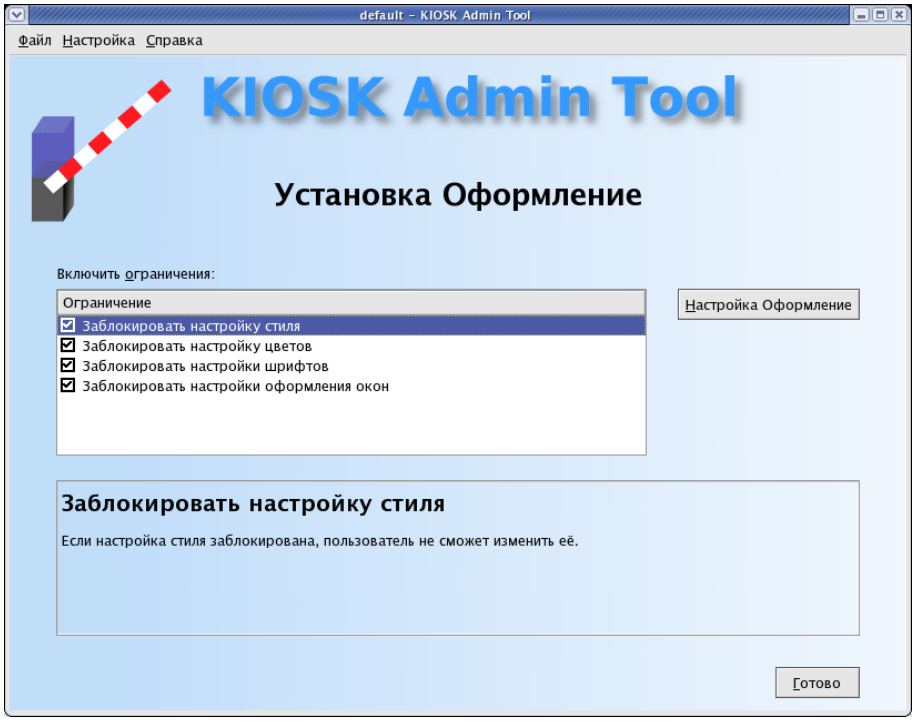

Рис. 9.4: Настройка конфигурации темы рабочего стола в Kiosk Admin Tool

  - Заблокировать настройки оформления окон (Lock down Window Decoration Settings)

- Настройка конфигурации KDE-панели
  - Заблокировать панель (Lock down panel)
  - Отключить контекстные меню (Disable Context Menus)
- Настройка прокси-сервера
  - Заблокировать настойки прокси-сервера (Lock down Proxy Settings)

В следующих двух секциях настолько много опций, что настройка их — это почти искусство. Мы рассмотрим эти опции более детально ниже в Разд. 9.1.4.

- Настройка конфигурации Konqueror
  - Отключить «Свойства» в контекстном меню (Disable Properties in context menu)
  - Отключить «Открыть с помощью» (Disable Open With action)
  - Отключить «Открыть в новой вкладке» (Open in New Tab action)
  - Отключить просмотр файлов вне домашнего каталога (Disable file browsing outside home directory)
- Настройка пунктов меню
  - Отключить Файл → Создать (Disable File → New)
  - Отключить Файл → Открыть (Disable File → Open)
  - Отключить Файл → Открыть недавние (Disable File → Open Recent)
  - Отключить Справка → О <Имя программы> (Help → About <Application>)
  - Отключить Справка → О KDE (Help → About KDE)

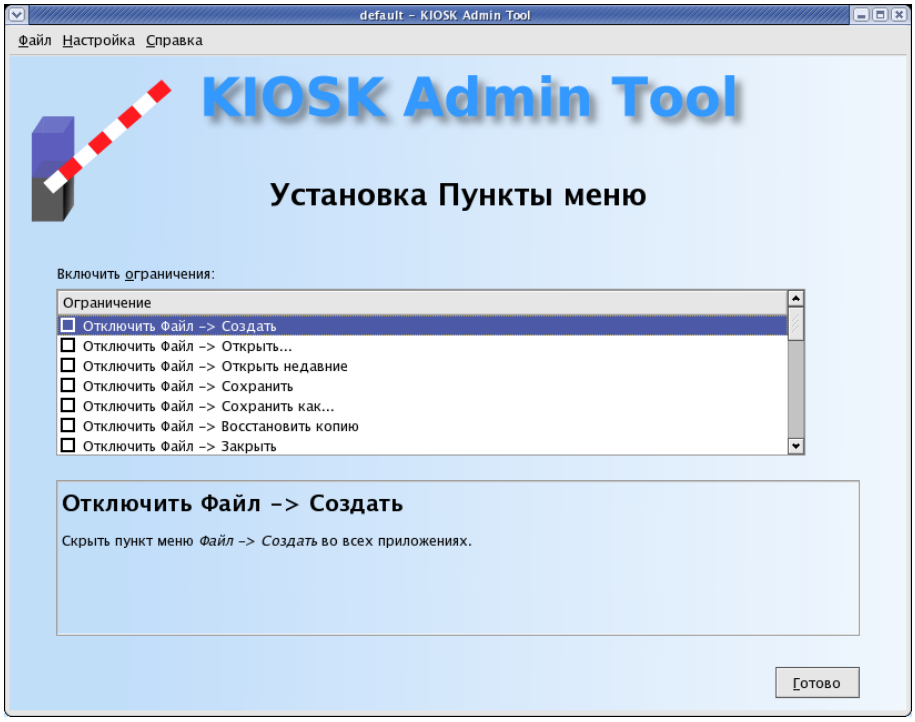

Рис. 9.5: Настройка ограничений на действия с помощью Kiosk Admin Tool

9.1.3 Неизменяемые элементы конфигурационных файлов

Как мы видели выше, создать профиль, ограничивающий действия пользователя, с помощью Kiosk Admin Tool достаточно просто. Но как будет работать механизм ограничения? Начиная с KDE3, конфигурационные опции могут быть помечены как неизменяемые. Это означает, что после того как значение такой опции установлено, оно не может быть изменено ни посредством KConfig, ни при помощи пользовательских настроек в `$KDEHOME` (обычно файл `$HOME/.kde`).

Как неизменяемые могут быть помечены отдельные элементы конфигурационного файла, группы элементов или все элементы, начиная с какого-то и до конца файла. Для этого перед элементом должен быть добавлен знак [$i] в подходящем месте (см. Прим. 9.1.3). Если KDE не имеет прав на запись по отношению к пользовательским конфигурационным файлам, они автоматически будут рассматриваться как неизменяемые, и пользователь получит предупреждение об этом факте. Если вы не хотите, чтобы предупреждение выводилось, добавьте опцию warn_unwritable_config=false в секцию ограничений на действия в KDE в файле `/etc/kderc` (или в файле `kdeglobals` на глобальном или пользовательском уровне, либо на уровне профиля). Как мы уже отмечали выше, просто запрет на запись в пользовательские конфигурационные файлы является недостаточным механизмом блокирования изменений, так как пользователь теоретически может переименовать эти файлы и создать новые с возможностью записи. Однако, ограничения, заложенные в операционную систему, дополняют более сложный и изощренный механизм блокирования, присущий среде KDE Kiosk.

9.1.4 Ограничения на действия

Рассматривая возможности Kiosk Admin Tool, мы уже перечислили те ограничения на действия, которые можно определить на уровне профиля. Эти изменения будут записаны в файле kdeglobals в секции KDE Action Restrictions. Ниже можно видеть, какого рода записи появились в конфигурационном файле после введения ограничений на действия для нашей ITSO группы.

Существует множество опций, которые можно добавить в эту секцию, в дальнейшем их число увеличится по мере увеличения числа приложений, использующих механизм KDE Kiosk. Вы можете использовать опции глобального уровня для ограничения действий меню и панели инструментов (например, action/file_new), а также для ограничения действий:

- Системы печати (print/options)

Пример 9.1.3: Блокирование элементов конфигурационного файла с использованием знака [$i]

```
[ScreenSaver]
Enabled[$i]=true

[Desktop0][$i]
Wallpaper=/usr/share/backgrounds/images/default.png
WallpaperMode=Scaled

[$i]
[Applet_1]
ConfigFile=kminipagerappletrc
DesktopFile=minipagerapplet.desktop
FreeSpace=0
WidthForHeightHint=92
```

Пример 9.1.4: Ограничения на действия в файле /etc/kde-profile/itso/share/config/kdeglobals

```
[KDE Action Restrictions][$i]
action/kdesktop_rmb=false
action/kicker_rmb=false
action/menuedit=false
editable_desktop_icons=false
editable_system_desktop_icons=false
movable_toolbars=false
run_command=fals
user/root=false
```

- Заставки экрана (opengl_screensavers)
- Рабочего окружения (logout, lock_screen, movable_toolbars, start_new_session)

Существуют также опции для ограничения действий стандартных KDE-программ:

- Konqueror (action/openintab) и KDesktop
- KWin (action/kwin_rmb)
- Kicker (action/kicker_rmb)
- Konsole (action/show_menubar)

Узнать, какие опции могут быть добавлены для того или иного приложения, можно с использованием команды dcop из командной строки или соответствующей программы графического окружения KDE. Например, после старта kmail можно ввести одну из приведенных ниже команд, для того, чтобы узнать какие опции могут соответствовать kmail.

```
% dcop kmail qt objects | grep KActionCollection | cut -d '/' -f 3

% dcop kmail kmail-mainwindow actions
```

Существуют опции, запрещающие запуск приложений, требующих определенных полномочий пользователя, например, опция user/root=false запрещает запуск всех приложений, которые требуют полномочий суперпользователя. Если вы интересуетесь тонкостями настройки ограничений на действия, обратитесь к руководству по KDE Kiosk на странице:

http://webcvs.kde.org/cgi-bin/cvsweb.cgi/kdelibs/kdecore/README.kiosk

9.1.5 Ограничения на URL

В среде KDE Kiosk можно ввести ограничения на действия, связанные с конкретным URL, а в некоторых случаях и ограничить доступ к URL. Как можно видеть в приведенном ниже примере, полный синтаксис таких ограничений довольно длинный, но мы остановимся на более простых случаях.

Пример 9.1.5: Ограничения на URL: полный синтаксис

```
[KDE URL Restrictions]
rule_count=<N->
rule_1=<act->,<ref_proto->,<ref_host->,<ref_path->,<proto->,/
<host->,<path->,<enabled->
...
rule_N=<act->,<ref_proto->,<ref_host->,<ref_path->,<proto->,/
<host->,<path->,<enabled->
```

Конкретные варианты использования этого механизмы вы можете видеть ниже. В первой части показано, как с помощью ограничения на действия запретить диалогу выбора файлов KDE выходить за пределы домашнего каталога. Первая строка запрещает просмотр каких-либо каталогов локальной файловой системы, вторая разрешает просмотр домашнего (`$HOME`) каталога. Вы могли заметить, что KDE раскрывает значение переменной окружения $HOME. Обычно мы должны помещать знак [$e] после имени переменной, или даже [$ei], для того чтобы предотвратить замену имени переменной окружения на ее актуальное значение, но в данном случае этого не требуется. Во второй части показано, как позволить пользователю открывать файлы в каталогах `$HOME` и `$TMP`, но запретить доступ к другим каталогам локальной файловой системы (файлы в интернете, тем не менее, остаются доступны).

Пример 9.1.6: Ограничения на URL: простые примеры

```
[KDE URL Restrictions][$i]
rule_count=2
rule_1=list,,,,file,,,false
rule_2=list,,,,file,,$HOME,true

[KDE URL Restrictions][$i]
rule_count=3
rule_1=open,,,,file,,,false
rule_2=open,,,,file,,$HOME,true
rule_3=open,,,,file,,$TMP,true
```

В конфигурационном файле KDE вы также можете использовать команды оболочки, как показано ниже, но мы не советуем злоупотреблять этой возможностью.

9.1.6 Ограничения на ресурсы

Последний вид ограничений, которые мы будем обсуждать — это ограничения на ресурсы KDE. Такое ограничение делает для пользователя невозможным подменить файлы, расположенные в каталоге `$KDEHOME/share` (где большинство приложений KDE держат свои ресурсы типа HTML-документации, звуковых файлов и т.д.) файлами, расположенными вне этого каталога. В первой части приведенного ниже примера мы запрещаем это для всех ресурсов, во второй вводится более тонкое ограничение: запрещается подмена звуковых файлов, фона и файлов локализации.

Вот полный список имен ресурсов, для которых можно вводить ограничения

- `date (share/data)`
- `html (share/doc/HTML)`

Пример 9.1.7: Общие ограничения на KDE-ресурсы

```
[KDE Resource Restrictions][$i]
all=fals

[KDE Resource Restrictions][$i]
sound=false
locale=false
wallpaper=false
```

- icon (share/icon)
- config (share/config)
- pixmap (share/pixmaps)
- apps (share/applnk)
- xdgdata-apps (share/applications)
- sound (share/sound)
- locale (share/locale)
- services (share/services)
- servicetypes (share/servicetypes)
- mime (share/mimelnk)
- wallpaper (share/wallpaper)
- templates (share/templates)
- exe (share/bin)
- lib (share/lib)
- all (share/)
- data_<имя приложения> (share/apps/<имя_приложения>)

Следует заметить, что имя каталога не всегда совпадает с именем элемента в конфигурационном файле, иногда из-за этого возникают ошибки. Ниже приведен пример, в котором показано, как использовать секции Control Module и Custom Restriction в файле `kdeglobals`.

Пример 9.1.8: Дополнительные опции ограничения на ресурсы KDE

```
[KDE Control Module Restrictions]
kde-background.desktop=false
kde-colors.desktop=false
kde-fonts.desktop=false
kde-kwindecoration.desktop=false
kde-proxy.desktop=false
kde-style.desktop=false

[KDE Custom Restrictions]
restrict_file_browsing=true
```

9.1.7 Ограничения в среде KDE Kiosk

Запрет запуска команды и доступа к командной оболочке могут быть весьма полезны, если речь идет об общедоступном терминале. Однако получить доступ к командному интерпретатору shell можно и при помощи броузера Mozilla, который не учитывает настройки среды Kiosk. Отключая возможность завершения сеанса, не забудьте также отключить возможность остановки X-сервера при помощи комбинации клавиш Ctrl+Alt+Backspace или перезагрузки машины при помощи комбинации Ctrl+Alt+Delete (отредактировав файл `/etc/inittab`), в противном случае ваши настройки окажутся совершенно бесполезны. Существует множество нюансов, которые необходимо учесть, чтобы обезопасить систему от несанкционированных действий, поскольку многие приложения и подсистемы Linux не учитывают настройки среды KDE Kiosk.

9.1.8 Подведем итоги

Мы надеемся, что после ознакомления с предыдущими разделами у вас появилось понимание того, какими возможностями обладает среда KDE Kiosk, и вы сможете эффективно использовать ее в предстоящей миграции на Linux-клиенты. Однако мы хотим предостеречь вас от излишнего увлечения конфигурированием KDE Kiosk. Чем больше вы меняете настройки, тем больше усилий вам потребуется для того, чтобы поддерживать их в дальнейшем. Если у вас есть опыт работы с консолью управления Microsoft (Microsoft Management Console, MMC), вам должно быть понятно, о чем идет речь. Профили Windows отличаются одной неприятной особенностью, их нельзя переносить покомпонентно, то есть вам придется полностью воспроизводить старые установки в новом окружении; KDE (и GNOME) предлагают очень широкие (и постоянно расширяющиеся) возможности выбора конфигурационных параметров.

9.2 Конфигурирование GNOME

В этом разделе мы покажем, как настроить рабочий стол GNOME для нужд работника, функции которого относятся к группе «Базовый Офис». Будет обеспечена работы приложений

- x3270
- Mozilla

Подробности того, как в GNOME организовано хранение данных пользовательского профиля можно найти также в Прил. В.

Структура каталогов будет соответствовать папкам, каталоги не будут размещены централизованно в одном месте, поскольку Linux — система многопользовательская, и каждый имеет свой домашний каталог. Каталог рабочего стола GNOME скрыт и имеет имя `$HOME/.gnome2`.

Мы будем работать со структурой рабочего стола, сохраняя данные в подкаталоге

```
$HOME/.gnome2/Business Applications
```

Работа с иконками организована также, как и в KDE. Это означает, что вы имеете дела с текстовыми файлами в формате ASCII. Вы можете сохранить их в каталоге `$HOME/.gnome2` для дальнейшего использования на рабочем столе.

9.2.1 Конфигурация в GNOME

Настройка конфигурации в GNOME — дело значительно более сложное по сравнению с настройкой конфигурации KDE, где для хранения конфигурации используются простые текстовые файлы в формате ASCII. Рабочий стол под управлением GNOME использует специальную базу данных, элементы которой хранятся в файлах в формате XML (Extensible Markup Language). Таких файлов множество и все они хранятся в каталоге `/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/`, если мы рассматриваем Novell Linux Desktop (для Red Hat Workstation в этом пути нужно заменить `/etc/opt/gnome` на `/etc`).

Ниже приведен список подкаталогов этого каталога, содержащих различные элементы настроек рабочего окружения GNOME:

```
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/accessibility
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/applications
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/background
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/file_views
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/font_rendering
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/interface
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/lockdown
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/peripherals
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/sound
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/thumbnailers
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/typing_break
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/url-handlers
```

Внутри каждого из этих подкаталогов находится файл с именем `%gconf.xml`. Ниже приведен пример такого файла

Пример 9.2.1: Файл %gconf.xml

```
/etc/opt/gnome/gconf/gconf.xml.defaults/desktop/gnome/lockdown
<?xml version="1.0"?>

<gconf>
   <entry name="disable_print_setup" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_print_setup"/>
   <entry name="disable_printing" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_printing"/>
   <entry name="disable_save_to_disk" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_save_to_disk"/>
   <entry name="disable_command_line" mtime="1089888947"
          schema="/schemas/desktop/gnome/lockdown/disable_command_line"/>
</gconf>
```

9.2.2 Инструмент конфигурирования gconftool-2

Менеджер рабочего стола GNOME снабжен инструментом, который называется gconftool-2. Вы можете вызывать его из командной строки, для того, чтобы вручную вносить соответствующие изменения в XML-файлы. Вы можете объявить некоторые значения обязательными (mandatory), тогда пользователи не смогут их изменить и именно эти значения будут использоваться при создании новых пользователей в системе.

ЗАМЕЧАНИЕ

Каждый из приведенных ниже примеров представляет собой командную строку. Эти примеры работают с GNOME2.6.

Эта команда меняет цвет фона на принятый по умолчанию и делает это значение обязательным:

```
gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.mandatory --type string --set
/desktop/gnome/background/picture_filename
```

```
/usr/share/backgrounds/images/corporate_logo.jpg
```

Второй пример ограничивает до двух число рабочих областей пользователя, таким образом пользователь может обновременно запускать только два приложения. В каждой рабочей области работает только одно приложение:

```
gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.mandatory --type int --set
/apps/metacity/general/num_workspaces 2
```

Число конфигурационных опций, которые контролирует gconftool-2, очень велико и мы не можем их все привести в этой книге, но вы можете набрать приведенную ниже команду и увидеть все параметры, установленные для текущего рабочего стола:

```
gconftool-2 -R /desktop
```

Хотя мы не можем охватить все опции gconftool-2, мы перечислим все опции, которые мы использовали для настроек своего рабочего стола и дадим пояснения по их настройке. Чтобы увидеть список всех опций, определенных для рабочего стола по умолчанию, наберите команду:

```
gconftool-2 -R /apps/panel
```

Можно вывести не все значения настроек, а только те, которые относятся к определенной области. Следующая команда выводит только значения глобальных для панели GNOME переменных:

```
gconftool-2 -R /apps/panel/global
```

Ниже приведен листинг, котороый выдает эта команда:

```
screenshot_key = Print
panel_hide_delay = 500
enable_animations = true
drawer_autoclose = true
keep_menus_in_memory = true
window_screenshot_key = <Alt>Print
panel_minimized_size = 3
tooltips_enabled = true
menu_key = <Alt>F1
confirm_panel_remove = true
panel_animation_speed = panel-speed-medium
highlight_launchers_on_mouseover = true
run_key = <Alt>F2
enable_key_bindings = true
panel_show_delay = 300
```

Эти значения можно редактировать при помощи gconftool-2. Например, приведенная ниже команда отключает run_key:

```
gconftool-2 --direct --config-source
xml:readwrite:/etc/gconf/gconf.xml.mandatory --type bool --set
/apps/panel/global/run_key false
```

Если мы заново просмотрим значения глобальных параметров, то увидим, что run_key отключен. Команда

```
gconftool-2 -R /apps/panel/global
```

выдает листинг:

```
screenshot_key = Print
panel_hide_delay = 500
enable_animations = true
drawer_autoclose = true
keep_menus_in_memory = true
window_screenshot_key = <Alt>Print
panel_minimized_size = 3
tooltips_enabled = true
menu_key = <Alt>F1
confirm_panel_remove = true
panel_animation_speed = panel-speed-medium
run_key = false
highlight_launchers_on_mouseover = true
enable_key_bindings = true
panel_show_delay = 300
```

Теперь run_key не доступен для всех пользователей, и они не могут изменить эту установку. Причина этого в том, что мы использовали каталог `gconf.xml.mandatory`. Можно вместо mandatory-каталога использовать каталог `gconf.xml.default`, тогда значение будет приписываться всем новым пользователям, но его можно будет впоследствии изменить.

Безусловно, gconftool-2 весьма полезный и удобный инструмент, однако существует инструмент с графическим интерфейсом, который называется gconf-editor. Этот инструмент обеспечивает графический интерфейс пользователя для доступа к GNOME XML-файлам, позволяет просматривать и изменять настройки. Он полезен, особенно при ознакомлении с содержанием различных XML-схем, которые используются при конфигурировании GNOME.

9.2.3 Ограничения в рабочем окружении пользователя

Работа с иконками в GNOME организована также, как и в KDE. Единственное отличие — это структура каталогов рабочего стола GNOME. Когда создается новый пользователь, то каталоги рабочего стола не создаются автоматически до того момента, пока он в первый раз не зарегистрируется в системе. Если же создавать эти каталоги вручную, то к ним не будут применены установки по умолчанию.

```
mkdir /home/fiona/.gnome2
```

```
mkdir /home/fiona/.gnome2/Business Applications
```

Структура каталогов для пользователя fiona должна быть следующая:

- `/home/fiona/.gnome2`
- `/home/fiona/.gnome2/Business Applications/x3270`
- `/home/fiona/.gnome2/Business Applications/mozilla`

Мы хотим, чтобы пользователь имел такие права доступа, чтобы мог запускать приложения, но не мог изменять их. Также мы хотим, чтобы он не имел возможности делать изменения на рабочем столе. Для этого изменим принадлежность каталога и права доступа:

```
chown -R <adminID> /home/<username>/.gnome2
```

```
chmod -R 755 /home/<username>/.gnome2
```

Такая установка позволяет пользователю запускать приложения, связанные с иконками на рабочем столе, но не позволяет делать в них изменения. Это не слишком надежный способ защиты, поскольку он не помешает

пользователю fiona переименовать каталог `/home/fiona/.gnome2` и создать другой рабочий каталог со своими свойствами.

9.2.4 Настройка внешнего вида рабочего стола

В этом разделе мы рассмотрим методы настройки внешнего вида рабочего стола GNOME с использованием графической программы *Центр Управления GNOME*.

> ЗАМЕЧАНИЕ
>
> В процессе создания персонализированного окружения для GNOME с использованием графического интерфейса в домашнем каталоге пользователя создается подмножество базы данных Gconf.

Чтобы открыть диалог для настроек, используйте в Nautilus URL

`preferences://`

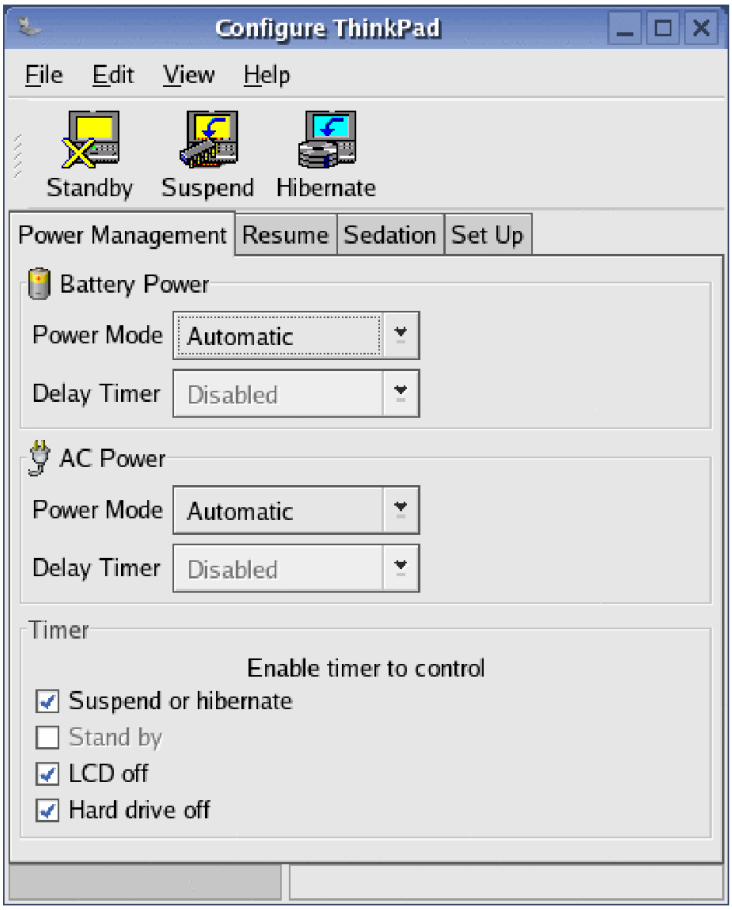

Рис. 9.6: Настройка Thinkpad в Центре управления GNOME

Другой путь — выбрать пункт Параметры в Главном меню или набрать команду **gnome-control-center** в командной строке. Центр управления GNOME позволяет просматривать огромное число настроек. Например, на самом верхнем уровне будет предложен следующий список установок:

- Дополнительные параметры
- Специальные возможности

- Фон рабочего стола
- Мышь
- Клавиатура
- Комбинации клавиш клавиатуры
- Меню и панели инструментов
- Шрифт
- Хранитель экрана
- Звук
- Тема
- Окна
- Разрешение экрана
- Удаленный рабочий стол
- Смена пароля
- Сменные накопители
- Сервис Прокси

Центр управления GNOME допускает расширение своих функций, то есть, в него можно добавить окно, которое будет отвечать за настойки внешнего вида нового приложения. На Рис. 9.6 вы можете видеть окно настойки программы Thinkpad, которое было добавлено в центр управления. Для этого был использован пакет configure-thinkpad, который можно загрузить со страницы в интернете по адресу:

http://tpctl.sourceforge.net/

9.3 Инструменты удаленного доступа

Модульная структура Linux обеспечивает очень широкие возможности для удаленной работы, особенно если речь идет о настольной машине, на которой установлен Тонкий или Стройный клиент. Эти возможности включают: базовый удаленный доступ, когда на мониторе рабочей станции фактически отображается рабочий стол удаленного компьютера, загрузку операционной системы по сети на бездисковую рабочую станцию, удаленное выполнение отдельного приложения на логическом дисплее, а также подключение нескольких пользователей одновременно к одной и той же рабочей станции.

9.3.1 Удаленный доступ

Базовый удаленный доступ состоит в том, что вся логика приложения запускается на сервере, а на рабочей станции остается только интерфейс пользователя. Примерами реализации такого подхода могут служить Virtual Network Computing (VNC), NoMachine NX и даже rdesktop, Linux-клиент для серверов RDP фирмы Microsof. Надо иметь в виду, что такой способ доступа требует значительных сетевых ресурсов, и поэтому его не используют в качестве стандартного для текущей работы, он применяется для удаленной диагностики, удаленной поддержки пользователя, при необходимости доступа на рабочую станцию с удаленного рабочего места.

9.3.2 Тонкий клиент

Если функции, выполняемые на рабочей станции, соответствуют типу Тонкий клиент, такая рабочая станция (это может быть бездисковая или обычная станция) допускает загрузку с удаленного сервера. В этом случае основные приложения запускаются на сервере, что облегчает их обновление и сопровождение. Пользовательские данные также хранятся на сервере, работа с ними происходит однотипно со всех рабочих станций. В зависимости от реализации этот тип доступа может требовать больше или меньше сетевых ресурсов, но в целом от сети требуется хорошая пропускная способность. Примерами Тонких клиентов могут служить Linux

Terminal Server Project (http://www.ltsp.org) и различные варианты Тонких клиентов фирмы Neoware, Inc. (http://www.neoware.com).

9.3.3 Перемещение приложений

Использование технологии клиент-сервер, принятой в системе X Window, открывает возможность для более избирательного подхода к удаленному доступу. Перемещение терминала на удаленный компьютер является естественным действием и не требует дополнительной работы, поскольку клиент системы X Window всегда взаимодействует с X-сервером посредством сетевых сокетов. Существует масса схем с использованием удаленного доступа к X-серверу, применение которых упрощает настройку системы и сокращает объем технической поддержки. Например, можно настроить конфигурацию так, чтобы почтовый клиент и веб-броузер стартовали локально, но запуск приложения, отвечающего за установку регулярных обновлений, происходил бы на удаленном сервере. Удаленное выполнение приложений X Window может осуществляться по протоколу SSH, что обеспечивает безопасную передачу данных во внешние сети. Можно использовать технологию сжатия NoMachine NX или FreeNX, это ускорит обмен данными и позволит работать по медленному соединению и даже по коммутируемой телефонной линии.

Подобный механизм удаленного выполнения может применяться и для Windows-приложений, для этого используются такие пакеты как Sun Secure Global Desktop Software (ранее известный как Tarantella) или Microsoft Terminal Server. Эти пакеты полезны в том случае, когда существуют приложения, не допускающие миграции на Linux.

9.3.4 Мультитерминальная обработка данных

Тонкие клиенты часто обладают низкой производительностью, большое число таких клиентов существенно повышает нагрузку на локальную сеть. Скорость выполнения приложений и время реакции зависят от пропускной способности локальной сети, в случае медленного соединения пользователи могут испытывать неудобства. Применение технологии мультитерминальной обработки данных на базе стандартных PC позволяет избежать этих проблем. Эта технология основана на применении видеокарт с несколькими входами и USB-устройств ввода данных. К одному компьютеру подключаются два или несколько комплектов монитор-клавиатура-мышь и несколько пользователей работают на одном компьютере одновременно. Как показывает практика, нагрузка на CPU и оперативную память в такой архитектуре возрастают незначительно.

Использование мультитерминальной обработки данных вместо предоставления каждому пользователю индивидуального компьютера сокращает расходы на приобретение оборудования и упрощает общую ИТ-инфраструктуру. Кроме того, сокращаются эксплуатационные расходы, упрощается техническая поддержка и сопровождение рабочей станции. Также сокращаются расходы на лицензии и уменьшается стоимость поддержки ПО сторонними организациями.

Практика использования мультитерминальной обработки данных сравнительно новая, но она перспективна, более подробно о ней можно прочитать, например, в издании *Linux Client Migration Cookbook, Practical Planning and Implementation Guide for Migrating to Desktop Linux*:

http://www.redbooks.ibm.com/redpieces/abstracts/sg246380.html

Глава 10

Сценарий миграции клиента

В этом разделе мы подробно рассмотрим пример миграции клиента. Сначала создадим описание клиента до миграции, выделим важные приложения, которые на нем функционируют. Затем составим план миграции в соответствии с требованиями, которые были сформулированы в Гл. 6 и Гл. 7. В последней части этой главы будут обсуждаться последовательные шаги и результаты миграции.

Эта глава содержит следующие разделы:

- Разд. 10.1 Пример миграции клиента
- Разд. 10.2 Детальный план миграции
- Разд. 10.3 Процесс миграции

10.1 Пример миграции клиента

В этом разделе мы пройдем миграцию одного клиента шаг за шагом, следуя методам, которые были описаны ранее. В качестве образца мы выбрали сравнительно простой тип клиента, но инфраструктура, в которую он включен, будет достаточно сложной. Начнем с описания клиента и общего планирования.

10.1.1 Описание паттернов использования клиента

Клиент, который мы использовали в качестве примера, является настольным компьютером, на котором разрабатывается техническая документация, точнее, работник входил в нашу группу ITSO. До миграции клиент обладал следующими свойствами:

- Это была рабочая станция в домене NT4
- Доступ к локальной сети и интернету осуществлялся с использованием Microsoft Internet Explorer
- Как средство создания форматированного текста применялся Adobe FrameMaker
- Для создания и редактирования образов экрана использовался Paint Shop Pro
- Печать шла через сетевой принтер
- Для обмена почтовыми сообщениями использовался Outlook в сочетании с Microsoft Exchange

На клиентской машине была установлена операционная система Windows 2000 с Service Pack 4. При миграции на Linux-клиент использовался дистрибутив Red Hat Desktop.

10.1.2 Определение наиболее существенных приложений и потребностей в интеграции

Поскольку основная функция сотрудника, использующего эту рабочую станцию, состояла в создании сложной по структуре документации, то основным используемым инструментом был Adobe FrameMaker. Сеть нужна

была, главным образом, для того, чтобы иметь возможность печатать на сетевом принтере и иметь доступ к общим файлам, размещенным в корпоративной сети.

Прежде всего заметим, что не существует версии Adobe FrameMaker, которая бы работала под Linux, то есть это приложение не является кросс-платформенным. Кроме того, оказалось, что использовать приложение с аналогичными функциями, но работающее под Linux, нельзя, поскольку это противоречило устоявшейся технологии. То есть, Adobe FrameMaker оказался приложением, не допускающим переноса на Linux. Мы будем работать с ним так, как было описано в Разд. 7.7.2.

Печать будет осуществляться по-прежнему через сетевой принтер, и будет организован доступ к файлам в домене NT4. Для этого будут использованы методы, описанные в Разд. 7.2.

10.2 Детальный план миграции

В этом разделе мы более подробно опишем, из каких шагов будет состоять миграция, обсудим проблемы, которые могут возникнуть на каждом этапе, посмотрим какими средствами поддерживается функциональная преемственность.

Выше мы описали рабочее место сотрудника ITSO. Теперь составим план работы по переводу этого рабочего места под управление Linux.

10.2.1 Какого типа этот клиент?

Рабочая станция сотрудника ITSO может быть отнесена к функциональной группе «Базовый офис» или «Стандартный офис», определенных в Разд. 6.1.1. Так как пользователь в основном занимается разработкой технической документации с использованием FrameMaker и при этом интенсивно пользуется веб-броузером для поиска необходимой информации в интернете, то по весу он принадлежит скорее к Толстым клиентам, если пользоваться терминологией Разд. 7.4.2.

10.2.2 Какое выбрать графическое окружение?

Важным вопросом является выбор графического окружения для нашего клиента. Варианты выбора обсуждались в Разд. 7.3.2.

Оба рассмотренных там варианта в целом удовлетворяют требованиям. Но в дистрибутивах Red Hat менеджером рабочего стола по умолчанию является GNOME, поэтому мы остановились на нем с целью упростить процедуру миграции. В дистрибутив Red Hat Desktop 4 включена версия GNOME 2.8.

10.2.3 Какое аппаратное обеспечение используется?

Для того чтобы убедиться в совместимости аппаратного обеспечения с новой операционной системой, составим соответствующую таблицу.

В нашем случае к компьютеру не подключено никакой периферии, печать и другие ресурсы доступны посредством локальной сети.

Как можно видеть, звуковая и видео карты интегрированы в материнскую плату, это означает, что для них могут потребоваться дополнительные драйверы. Мы обратились на страницу Intel в сети и скачали оттуда драйверы для этих компонент, про остальные элементы конфигурации было известно, что они стандартны и не могут вызвать проблем.

10.2.4 План замены программного обеспечения

Ниже в форме таблицы показано, какие приложения будут обеспечивать функциональность нашего рабочего места после перехода на Linux.

Таблица 10.1: Описание аппаратного обеспечения

Компоненты	Описание
Модель PC	IBM Netvista Type 6759
Процессор	Intel Pentium® III 866 MHz
Чипсет	i810 integrated
Видео-карта	i810
Звуковая карта	i810
Устройство чтения CD	CD-ROM 24x IDE
Флоппи-дисковод	3,5"floppy drive
Жесткий диск	IDE Harddisk 40 GB

Таблица 10.2: Схема замены приложений при миграции клиента

Приложение под Windows	Приложение под Linux
Microsoft Internet Explorer	Mozilla Firefox
Microsoft Outlook	Novell Evolution с Novell Connector для Exchange
Windows Explorer	Nautilus
WinZip	FileRoller
ICQ	Gaim
Microsoft Word	OpenOffice.org Writer
Microsoft Excel	OpenOffice.org Calc
Microsoft Powerpoint	OpenOffice.org Impress
Adobe Reader	Acrobat Reader для Linux
Adobe FrameMaker	Adobe FrameMaker (на Windows Terminal Services)
Jasc Paint Shop Pro	GIMP

10.2.5 Подключение к сети Windows

Подключение к сети Windows в некоторых случаях может быть сложной задачей, например, если используется удаленное хранение профилей, или структуры Active Directory, или комбинация нескольких доменов с опцией «доверительный». В нашем случае не было ни одного из этих осложняющих факторов.

Для нашего клиента нужен был доступ к общим сетевым каталогам файл-сервера домена Windows и сетевая печать через сервер печати, также включенный в домен Windows. Поэтому нужно было имя домена и существующий пользователь для авторизации в домене ITSO.

В нашем случае использовались имена:

- Domain name = ITSOAUSNT
- Primary domain controller = ITSONT00
- Backup domain controllers = ITSONT01,ITSONT02,ITSONT03
- User name(s) for residents = ausresxx

Далее для интеграции Linux-клиента в локальную сеть были использованы методы, подробно описанные в Разд. 7.2 и Гл. 11.

10.3 Процесс миграции

В этом разделе обсуждается процедура выполнения миграции.

10.3.1 Выполнение базовой инсталляции

Первым шагом является базовая установка операционной системы с дистрибутива. В нашем случае использовался дистрибутив Red Hat Desktop, представляющий собой набор из четырех CD. Некоторые специальные пакеты были получены из интернета с использованием RHN. А именно, мы скачали пакеты, содержащие ПО независимых производителей, такое как Acrobat Reader и Real Player, а также набор высококачественных шрифтов truetype производства фирмы AGFA.

Во время установки мы приняли предложенный по умолчанию инсталляционной программой выбор пакетов прикладных программ. Программа установки Anaconda фирмы Red Hat отработала на нашем оборудовании без каких-либо проблем. Программа тестирования аппаратного обеспечения, kudzu, также отработала успешно, все компоненты были определены правильно. Нужные модули для видео и звуковой карты были загружены. Также был загружен драйвер сетевой карты и в конце установки был подключен доступ к сети, при этом система успешно получила необходимую информацию от DHCP-сервера в сети ITSO. Сервис разрешения имен отработал нормально, без каких-либо дополнительных настроек.

10.3.1.1 Установление удаленного доступа администратора с использованием VNC

После того как установка была завершена, мы изменили настройки на нашем клиенте, чтобы иметь возможность удаленного администрирования с использованием VNC. Сначала следует проверить, что ssh демон запущен, и что посредством VNC будет доступно полноценное рабочее окружение. Для этого, войдя с полномочиями суперпользователя, надо внести изменения в файл `$HOME/.vnc/xstartup` как показано в примере приведенном ниже.

Пример 10.3.1: Подключение VNC-соединения

```
#!/bin/sh

# Uncomment the following two lines for normal desktop:
unset SESSION_MANAGER
exec /etc/X11/xinit/xinitrc
.....
```

Также необходимо включить Xdmcp в файле `/etc/X11/gdm/gdm.conf` и задать установки сессии в файле `/etc/sysconfig/vncservers`. VNC-сервер запускается командой

service vncserver start

После этого вы можете соединиться с любым VNC клиентом и работать с использованием графического интерфейса.

> **ВАЖНО**
>
> VNC — это мощное средство, его надо использовать с осторожностью. Оно позволяет использовать локальный X-сервер практически из любого места локальной сети. Можно ограничить доступ к рабочей станции, внеся соответствующие изменения в файлы `/etc/hosts.allow` или `/etc/hosts.deny`.

10.3.1.2 Подготовка к интеграции с сетевыми сервисами Windows

Следующим шагом нашей миграции будет подключение к сетевым сервисам Windows. В нашем случае имеются серверы под управлением Windows NT и Windows 2000, поэтому мы может следовать инструкциям, изложенным в Гл. 11.

Во-первых, нам надо подключиться к Windows-домену, то есть необходимо обеспечить авторизацию в этом домене. Для того чтобы войти с Linux-клиента с использованием имени и пароля, определенных в Windows-домене, надо включить **winbind**.

Ниже показано, какие изменения для этого надо внести в файл `/etc/samba/smb.conf`.

Пример 10.3.2: Изменения, вносимые в файл smb.conf для запуска winbind

```
[global]
    workgroup = ITSOAUSNT
    security = domain
    password server = ITSONT00,ITSONT01,ITSONT02,ITSONT03
...
...
    winbind separator = +
    idmap uid = 10000-20000
    idmap gid = 10000-20000
    winbind enum users = yes
    winbind enum groups = yes
    template homedir = /home/%D+%U
    template shell = /bin/bash
```

10.3.2 Интеграция с существующими сетевыми сервисами

Итак, на стадии подготовки мы запустили на клиентской стороне демон **winbind**, поэтому набрав команду **wbinfo -u**, мы должны увидеть список пользователей домена Windows. Но для того чтобы это сработало, наш Linux-клиент должен быть зарегистрирован в домене Windows, поскольку до тех пор пока для клиента нет учетной записи, он не может получить доступ через **winbind**. Для того чтобы зарегистрироваться в домене, нужно выполнить команду:

```
net join -a ITSOAUSNT -U administrator
```

> ВАЖНО
>
> Не забудьте зарегистрироваться в домене, общий формат команды **net join -a <domainname>**. Эта команда создает учетную запись на контроллере домена и обеспечивает Linux-клиенту SID для авторизации.

Теперь мы сможем прочесть список пользователей, и осталось выполнить еще два шага для дого, чтобы мы могли использовать учетную запись Windows для входа на Linux-клиент.

Сначала внесем изменения в файл `nsswitch.conf`, что позволит установить соответствие между пользователями и группами Windows и пользователями Linux (подробнее см. в Разд. 11.2).

Пример 10.3.3: Изменения в фйле etc/nsswitch.conf

```
passwd: files winbind
group: files winbind
```

Для проверки можно набрать команду **getent passwd**, должен появиться список всех пользователей с корректными идентификаторами uid и gid.

Теперь для того чтобы процесс авторизации заработал, необходимо задействовать два PAM-модуля (подробнее об этом в Разд. 11.4.1).

Так как мы используем для нашей пилотной миграции дистрибутив Red Hat Desktop, то в нашем случае надо взять информацию из Разд. 11.4.1.1.

Мы должны включить модули `pam_winbind` и `pam_smb_auth`, которые обеспечат пользователям возможность авторизации в домене ITSOAUSNT. Для этого отредактируем файл `/etc/pam.d/system_auth`, как показано ниже

Пример 10.3.4: Изменения в файле /etc/pam.d/system_auth

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth    required   /lib/security/$ISA/pam_env.so
auth    sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
auth    sufficient /lib/security/$ISA/pam_winbind.so use_first_pass
auth    sufficient /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth    required   /lib/security/$ISA/pam_deny.so

account required   /lib/security/$ISA/pam_unix.so
account sufficient /lib/security/$ISA/pam_winbind.so
.......................
```

Теперь стало возможно входить на Linux-клиент как с локальными, так и с доменными именами пользователя. Первая проба, вход с консоли, показывает, что все в порядке. Однако попытка графической авторизации через приглашение GNOME генерирует ошибку.

Причина этого в том, что до первого графического входа в систему для пользователя еще не создано домашнего каталога, и GNOME пытается использовать системные установки для создания файлов пользовательских настроек. Для того чтобы решить эту проблему, надо использовать еще один PAM-модуль, `pam_mkhomedir`. При помощи этого модуля во время авторизации создаются домашние каталоги пользователя, если они еще не существуют.

Итак, добавляем следующие строки в файл `/etc/pam.d/system_auth`:

Пример 10.3.5: Добавление модуля pam_mkhomedir в файл /etc/pam.d/system-auth

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is
auth    required   /lib/security/$ISA/pam_env.so
auth    sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
auth    sufficient /lib/security/$ISA/pam_winbind.so use_first_pass
auth    sufficient /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth    required   /lib/security/$ISA/pam_deny.so

account required   /lib/security/$ISA/pam_unix.so
account sufficient /lib/security/$ISA/pam_winbind.so
...................
session optional  /lib/security/$ISA/pam_mkhomedir skel=/etc/skel umask=0022
```

Теперь графическая авторизация через приглашение GNOME проходит успешно, в процессе авторизации создаются домашние каталоги.

Для входа в систему мы использовали ID:

ITSOAUSNT+AUSRES06

В общем виде формат имени для регистрации выглядит так:

<имя_домена:разделитель:имя_пользователя>

В нашем случае в качестве разделителя был использован знак <+>.

К сожалению, в стандартной установке GNOME нет возможности ссылаться на ID пользователя без доменного имени, например, иконка домашнего каталога пользователя будет иметь название **ITSOAUSNT+AUSRES06**.

10.3.2.1 Монтирование общедоступных каталогов Windows

Кроме регистрации в домене Windows, нам надо еще обеспечить доступ к файлам, хранящимся в общедоступных каталогах, принадлежащих этому домену. В нашем случае пользователю надо иметь доступ к трем таким каталогам. Мы сохраним информацию, необходимую для авторизации, в файлах и будем использовать ее на этапе монтирования ресурсов. Команда **smbmount** допускает монтирование удаленных каталогов с использованием учетной записи пользователя, хранящейся в специальном файле. По соображениям безопасности этот файл будет расположен в домашнем каталоге суперпользователя и доступ к нему будет разрешен только суперпользователю. Ниже показан пример файла /root/.credentials с параметрами пользователя.

Пример 10.3.6: Файл /root/.credentials с параметрами пользователя для монтирования удаленных каталогов

```
username = ausres06
password = password
domain = itsoausnt
```

Правильным подходом будет включить процедуру монтирования удаленных каталогов в /etc/fstab, с тем чтобы удаленные каталоги монтировались одновременно с монтированием локальных файловых систем. Ниже приведен пример монтирования каталогов с файловой системой типа smbfs в файле /etc/fstab.

Пример 10.3.7: Монтирование удаленых каталогов в /etc/fstab

```
/dev/hda1     /          reiserfs   acl,user_xattr        11
/dev/hda5     /scr           auto   noauto,user           00
/dev/hda2     swap           swap   pri=42                00
......
/dev/fd0      /media/floppy subfs  fs=floppyfss,procuid,nodev,nosuid,sync 0 0
//itsont05/data /shares/data_g smbfs credentials=/root/.credentials 0 0
```

Этот прием хорошо работает во всех случаях, кроме тех, когда в именах общедоступных каталогов Windows используются специальные символы. В противоположность UNIX-подобным системам, Windows позволяет использовать в именах разделяемых файловых ресурсов пробельные символы и символы, не входящие в стандартный набор ASCII. Например, имя каталога может содержать умляут, в подобных случаях при удаленном монтировании возникнут проблемы. Проблему также представляют собой пробелы в именах каталогов. В файле /etc/fstab пробелы воспринимаются как разделители, поэтому попытка смонтировать каталог с именем «project data» вызовет несколько сообщений об ошибке в файле /etc/fstab, поскольку «date» будет воспринята как точка монтирования.

ПОДСКАЗКА

При монтировании общедоступных каталогов Windows следует проверить, удовлетворяют ли их имена стандартам, принятым в UNIX-подобных системах, и, если не удовлетворяют, то по возможности переименовать их.

Поскольку нам обязательно надо было смонтировать каталог с именем «project data», мы воспользовались другим приемом, а именно, включили команду **smbmount** в файл `/etc/rc.local`, как показано ниже.

Пример 10.3.8: Пример монтирования удаленных каталогов в файле /etc/rc.local

```
#!/bin/sh
#
# This script will be executed *after* all the other init scripts.
# You can put your own initialization stuff in here if you don't
# want to do the full Sys V style init stuff.
touch /var/lock/subsys/local
#mounting windows file shares
smbmount //itsont05/data /shares/data_g -o credentials=/root/.credentials, \
 gid=ITSOAUSNT+Domain\ Users,dmask=0775
smbmount //itsont02/"project data" /shares/projectdata_e \
 -o credentials=/root/.credentials,gid=ITSOAUSNT+Domain\ Users,dmask=0775
```

Как показано в последней строчке примера, имя каталога с пробелом помещено в кавычки, так же следует поступать и со специальными символами. Опции gid и dmask команды **smbmount** использованы для того, чтобы Linux-каталоги, которые являются точками монтирования удаленных каталогов, имели соответствующие права доступа. Если бы они были пропущены, то пользователи не имели бы прав на запись, так как каталоги были бы смонтированы со стандартными параметрами для каталога пользователя root. Права доступа к каталогу на стороне Windows-сервера контролируются средствами Windows после того, как каталоги смонтированы. Ясно, что права, установленные на серверной стороне, имеют приоритет по сравнению с правами, прописанными в smbmount.

ВАЖНО

Будьте внимательны при установлении прав доступа к Linux-каталогам, которые будут использоваться как точки монтирования общедоступных каталогов Windows-сервера. Используйте параметр gid, для того чтобы предотвратить доступ к этим каталогам пользователей, не являющихся членами авторизированной группы.

10.3.2.2 Организация печати

Сетевую печать можно организовать также с использованием протокола smb, но мы выбрали использование системы печати CUPS, поскольку это более простой путь.

> **Подсказка**
>
> Если драйвер к принтеру, который вы собираетесь использовать, отсутствует в стандартном наборе, попытайтесь найти файл с драйвером в сети. Если он обнаружится, то поместите нужный ppd-файл (Postscript Printer Description) в каталог `/usr/share/cups/model`. Затем нужно перезапустить CUPS и добавить принтер. Искать драйвер имеет смысл в следующем порядке:
>
> - Обратитесь к веб-станице производителя принтера
> - Посмотрите, нет ли нужного файла на странице ПО печати под Linux:
>
>   http://www.easysw.com/printpro
> - Попробуйте использовать фалы PPD, поставляемые фирмой Adobe:
>
>   http://www.adobe.com/products/printerdrivers/winppd.html
> - На домашней странице CUPS можно найти много полезных ссылок:
>
>   http://www.cups.org

Подключение принтеров с использование системы печати CUPS — очень простая процедура, она описана Разд. 11.7.2.

10.3.3 Установка и настройка приложений

Как уже было отмечено выше, одним из основных приложений на выбранной рабочей станции является Adobe FrameMaker. Некоторое время назад Adobe предлагала версию этого продукта для Linux, но эта практика прекратилась после выхода версии 6.0. Так как нашему сотруднику нужны были специфические плагины, то требовалось работать именно с Adobe FrameMaker 7.0. Таким образом, данное приложение оказалось немигрируемым.

Для обеспечения работы Linux-пользователей, которым необходим Adobe FrameMaker, было решено установить приложение на Windows-сервере и организовать удаленный доступ посредством WTS (Windows Terminal Server). Эта технология позволяет Linux-клиентам соединяться по протоколу RDP (Remote Display Protocol) с приложениями, запущенными на WTS-сервере. Существуют и другие терминальные сервисы для доступа к приложениям, запущенным на удаленном сервере, но мы предположили, что для наших нужд будет достаточно WTS.

Надо иметь в виду, что нет никакой гарантии, что любое приложение можно запустить на терминальном сервере. Поэтому мы провели серию тестов, запуская FrameMaker несколько раз из разных сессий. Мы обнаружили, что наша версия FrameMaker не оптимизирована для работы по такому сценарию, так как каждый запущенный на WTS-сервере экземпляр приложения забирал такое же количество оперативной памяти, как и первый.

> **Важно**
>
> Если приложение не оптимизировано для работы на терминальном сервере, запуск каждого нового экземпляра приложения будет забирать столько же оперативной памяти, сколько и запуск первого. Поэтому терминальный сервер должен иметь достаточно оперативной памяти для одновременного запуска всех ожидаемых экземпляров данного приложения.

Соединение с терминальным сервером обеспечивается приложением-клиентом с графическим интерфейсом tsclient. Для хранения конфигурационных данных tsclient используются файлы `$HOME/*.tc`. Синтаксис для запуска сессии tsclient будет следующий:

```
tsclient -x $HOME/framemaker.tc
```

Файл `framemaker.tc` содержит всю необходимую информацию для соединения: IP-адрес сервера, имя пользователя и т.д.

В следующем разделе на Рис. 10.1 показан пример экрана, на котором FrameMaker запущен в терминальной сессии с Linux-клиента.

10.3.3.1 Работа с изображениями

На Winows-клиенте для работы с изображениями и создания образов экрана использовалась программа Jasc Paint Shop Pro. Для этой программы нет версии, работающей под Linux, но есть функциональный аналог GIMP. GIMP — это мощная программа с открытым кодом, в нашем случае ее функциональные возможности полностью покрывали все нужды пользователей. Подробную информацию о GIMP можно найти в сети по адресу: http://www.gimp.org.

GIMP включен как стандартный пакет в дистрибутив Red Hat Desktop, таким образом специально устанавливать и настраивать его не требуется.

В следующем разделе на Рис. 10.2 показан пример экрана, на котором запущен GIMP.

10.3.3.2 Броузер

В нашем сценарии миграции в качестве базового броузера был использован Mozilla Firefox. Firefox предоставляет только возможности броузера, почтовый клиент в него не интегрирован, но нас такая ситуация вполне устраивала, так как в качестве почтового клиента планировалось использовать Novell Evolution.

Следует иметь в виду, что некоторые веб-сайты требуют специальных возможностей броузера (например, сайты, использующие ActiveX® controls или специальные функции JavaScript™) и могут быть просмотрены только с использованием броузера Microsoft Internet Explorer. Проверьте список важных для пользователя адресов, прежде чем выбирать веб-клиента. В нашем случае доступ к подобным сайтам не требовался, поэтому не было никаких препятствий для использования в качестве броузера Mozilla Firefox.

10.3.3.3 Почтовый клиент

Чтобы сделать наш сценарий миграции более реалистичным, добавим требование, что существующие клиенты используют Microsoft Outlook, который соединяется с сервером Microsoft Exchange 2000. Таким образом, необходимо выполнить замену Windows-клиента Outlook на функционально аналогичный Linux-клиент и обеспечить его интеграцию в существующую систему обмена сообщениями на базе сервера Exchange 2000.

Использование Novell Evolution в сочетании с Novell Connector for Exchange. Для решения этой задачи был использован почтовый клиент Novell Evolution вместе с Novell Connector в качестве инструмента доступа к серверу Microsoft Exchange. Почтовый клиент Evolution построен таким образом, что допускает использование специальных способов соединения с различными программными средствами коллективной работы. Коннектор для Microsoft Exchange 2000 и 2003 был выпущен под лицензией GPL в мае 2004 года. Ключевым фактором для возможности его разработки было включение фирмой Microsoft в сервер Exchange веб-интерфейса, котрый был назван OWA (Outlook Web Acces). В OWA используется WebDAV для доступа к месту хранения сообщений на сервере. Соответственно, стало возможно обмениваться сообщениями с Exchange, используя интерфейс WebDAV, вместо того чтобы заниматься обратным инжинирингом протокола MAPI.

> ВНИМАНИЕ
>
> Сервер Exchange поддерживает интерфейс WebDAV только в версиях 2000 и 2003. В момент написания этой книги не существовало коннектора, обеспечивающего связь Linux-клиента с сервером Exchange версии 5.5.

Мы установили необходимые для работы Novell Connector for Microsoft Exchange Server пакеты RPM дистрибутива Red Hat Desktop и убедились, что с их помощью можно легко установить соединение с сервером Exchange. Чтобы это работало, должны быть выполнены некоторые условия. Главное, на сервере должен быть присутствовать Outlook Web Access.

> ВНИМАНИЕ
>
> Для того, чтобы можно было использовать коннектор Novell Connector для сервера Exchange, нужно проделать некоторые предварительные действия. Во-первых, должен быть запущен виртуальный сервер Exchange в HTTP-секции менеджера системы Exchange. Во-вторых, для пользователей должен быть доступен протокол HTTP, для этого надо сделать соответствующие настройки посредством консоли ADU (Active Directory Users).

Дополнительную информацию о Novell Connector for Microsoft Exchange Server можно найти в сети по адресу:

http://www.novell.com/products/evolution

10.3.3.4 Перенос пользовательских настроек

Для того чтобы облегчить нашим сотрудникам адаптацию к новой системе, мы решили перенести пользовательские настройки и данные. Согласно принятому порядку пользовательские документы сохраняются в сетевых каталогах общего доступа, поэтому нам не приходилось заботиться об их поиске на диске. Предполагая, что какие-либо заметки или важные черновики могли быть сохранены локально, мы решили перенести содержимое каталогов `My Documents` и `Desktop`. Почтовые сообщения пользователей хранятся централизовано, на сервере Exchange Server, поэтому нужно было переместить только информацию об учетной записи. Для того чтобы пользователям было комфортно работать с новым веб-броузером, мы решили перенести в новое окружение их закладки и стартовые страницы.

Перенос пользовательских настроек вручную. Сначала рассмотрим, как перенести все эти пользовательские данные вручную. Windows-версия броузера Firefox умеет импортировать такие пользовательские данные, как закладки и стартовые страницы из настроек Microsoft Internet Explorer, а Linux-версия этой опцией не обладает. Поэтому мы установили Firefox на машине с ОС Windows и выполнили импорт настроек Microsoft Internet Explorer в Firefox. Затем мы открыли общий доступ к папкам `My Documents` и `Desktop`, а также к папке, в которой хранятся настройки Firefox (обычно это папка `Application Data\Mozilla`). В нашем случае настройки хранились в файле

```
C:\Documents and Settings\AUSRES06\Application Data
    \Mozilla\Firefox\Profiles\xyjib4ms.default
```

Затем мы запустили почтовую программу Microsoft Outlook и убедились, что все почтовые сообщения хранятся на сервере. Если бы обнаружилось, что некоторые почтовые папки размещены локально, то мы создали бы каталог на сервере и скопировали бы их. Этот шаг был последним на машине под Windows.

На Linux-клиенте мы запустили Nautilus и получили доступ к общим ресурсам Windows. Мы скопировали содержимое каталога `My Documents` в подкаталог `Documents` домашнего каталога пользователя. Все файлы каталога `Desktop` (исключая файлы ярлыков) были скопированы на рабочий стол. Затем мы запустили Firefox, чтобы создать новый профиль пользователя. Файлы профиля хранятся в подкаталоге `.mozilla` домашнего каталога пользователя. На нашей машине полный путь к каталогу профиля был такой:

```
/home/ITSOAUSNT+AUSRES06/.mozilla/firefox/xdch2a44.default
```

Затем мы скопировали все файлы старого профиля с Windows-машины в этот каталог поверх тех фалов, которые в нем были. И, наконец, мы запустили Evolution и создали новую учетную запись пользователя на Exchange Server.

Автоматизированный перенос пользовательских настроек. Поскольку у нас было несколько рабочих станций с однотипным рабочим окружением, мы решили автоматизировать процесс переноса пользовательских данных. Для этого использовался инструмент Progression Desktop. Подробную информации о Progression Desktop можно найти на странице http://www.versora.com. В наши планы входило выполнить миграцию более чем на 10-ти рабочих станциях, поэтому мы решили использовать для автоматизации всех процессов интерфейс командной строки. Мы написали скрипты, которые позволили нам свести всю процедуру переноса пользовательских данных к двум простым шагам.

Во-первых, мы создали шаблон, в который включили все данные, которые предполагалось переносить, и сохранили его в общедоступном каталоге в сети. Затем, мы написали `.bat`-файл для автоматической установки .NET framework и Progression Desktop и сохранения настроек в пакете PNP (Platform Neutral Package) на общедоступном каталоге в сети.

Пример 10.3.9: Пакетный файл для установки и запуска Progression Desktop под Windows

```
cmd /c \\itsont05\data\ProgressionDesktop\Windows\dotnetfx.exe /q:a
      /c:"install /q /l"
cmd /c msiexec /quiet /i
      \\itsont05\data\ProgressionDesktop\Windows\zf7fg42j9.msi
"c:\Program Files\Versora\Progression Desktop\ProgressionDesktopCore.exe"
      --store
      --file=\\itsont05\data\%UserDomain%+%UserName%.pnp
      --template=\\itsont05\data\ITSOtemplate.xml
```

Затем мы написали shell-скрипт для машины под Linux, который временно устанавливал на компьютере Mono (наш дистрибутив не включал Mono), запускал Progression Desktop для импорта пользовательских настроек из PNP-пакета (Platform Neutral Package) с сетевого каталога и после окончания работы удалял пакет Mono.

Пример 10.3.10: Скрипт для установки и запуска Progression Desktop под Linux

```
#! /bin/bash
/shares/data_g/monoinstaller/mono-1.1.8.2_1-installer.bin \
      --mode unattended --prefix /var/tmp/mono --AddMonoToPath 0
/var/tmp/mono/bin/mono \
      /shares/data_g/ProgressionDesktop/Linux/ProgressionDesktopCore.exe \
      --apply --file=/shares/data_g/$USER.pnp \
      --template=/shares/data_g/ITSOtemplate.xml
rm -rf /var/tmp/mono
rm -f ~/Desktop/Mono.desktop
```

Теперь для переноса пользовательских данных с машины, работающей под Windows на машину под ОС Linux нужно было выполнить два действия: сначала запустить `.bat`-файл на первой, а после того как он отработает, запустить shell-скрипт на второй.

10.3.3.5 Пользовательские настройки GNOME

Существует масса возможностей для изменения внешнего вида и функциональности рабочего стола Linux-клиента. Нас, в общем, устраивал внешний вид стандартного рабочего стола GNOME, но мы решили изменить некоторые настройки, чтобы показать, как это делается.

Прежде всего мы решили подобрать подходящий цвет для фона рабочего стола и поместить на него в логотип нашей группы. Это легко можно сделать из меню GNOME или вызвав в командной строке gconftool-2 как показано ниже.

Более подробное описание процедуры конфигурирования GNOME и использования gconftool-2 вы найдете в Разд. 9.2.

Пример 10.3.11: Использование gconftool-2 для настройки фона экрана

```
# gconftool-2 --direct --config-source \
     xml:readwrite:/etc/gconf/gconf.xml.mandatory --type string --set \
     /desktop/gnome/background/picture_filename itso-logo.png
```

Кроме того, можно использовать другой логотип для кнопки главного меню. Red Hat заменяет логотип GNOME своим специально созданным изображением. Мы заменили его на другое, отредактировав файл

`/usr/share/pixmaps/redhat-main-menu.png`

Имя этого файла прописано в GConf. Мы избрали самый простой путь и просто переименовали файл, содержащий логотип группы ITSO, в `redhat-main-menu.png`. Естественно, этот файл должен иметь подходящий размер и разрешение, в нашем случае он был 48x48 пикселов и при помощи GIMP был сделан прозрачным.

И последнее изменение на рабочем столе состояло в том, что мы вывели на него ссылки на каталоги, расположенные на Windows-сервере, для удобства пользователей их назвали теми же буквами, какие использовались в старом клиенте.

Результат наших настроек можно увидеть ниже на Рис. 10.5.

10.3.4 Образы экранов: миграция клиента на Linux

Ниже мы собрали вместе образы экранов, демонстрирующие работу основных приложений на Linux-клиенте.

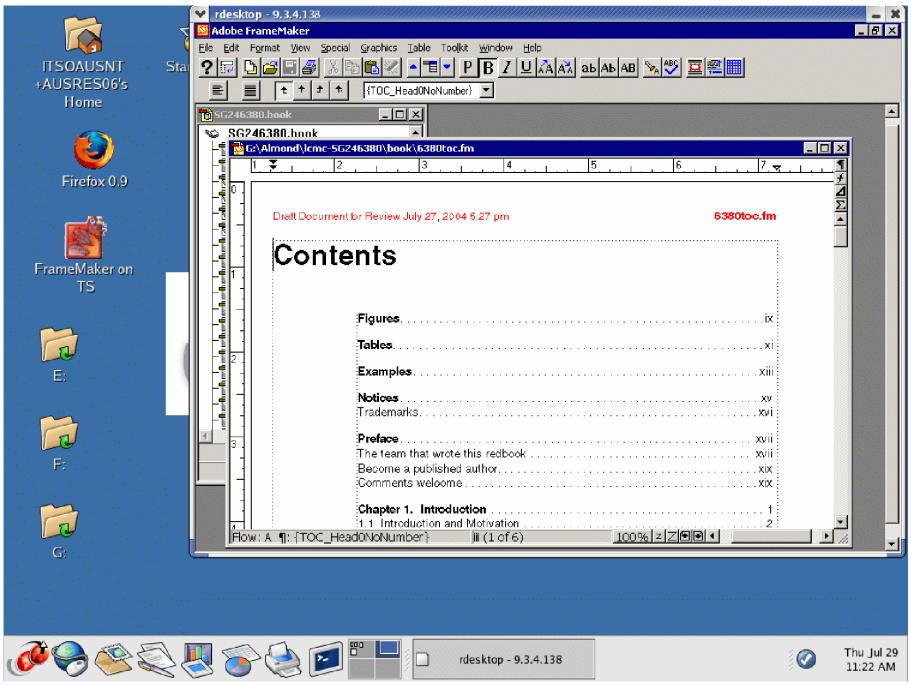

Рис. 10.1: Работа FrameMaker на Windows-сервере в терминальной сессии, запущенной с Linux-клиента (образ экрана публикуется с разрешения корпорации Adobe Systems)

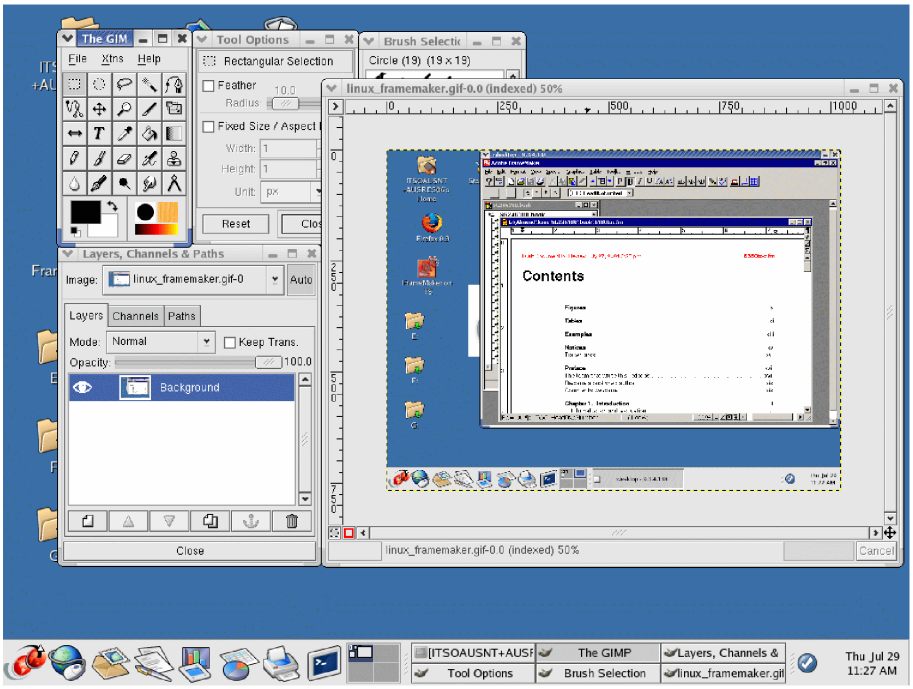

Рис. 10.2: Использование Gimp для получения образа экрана с сессией FrameMaker

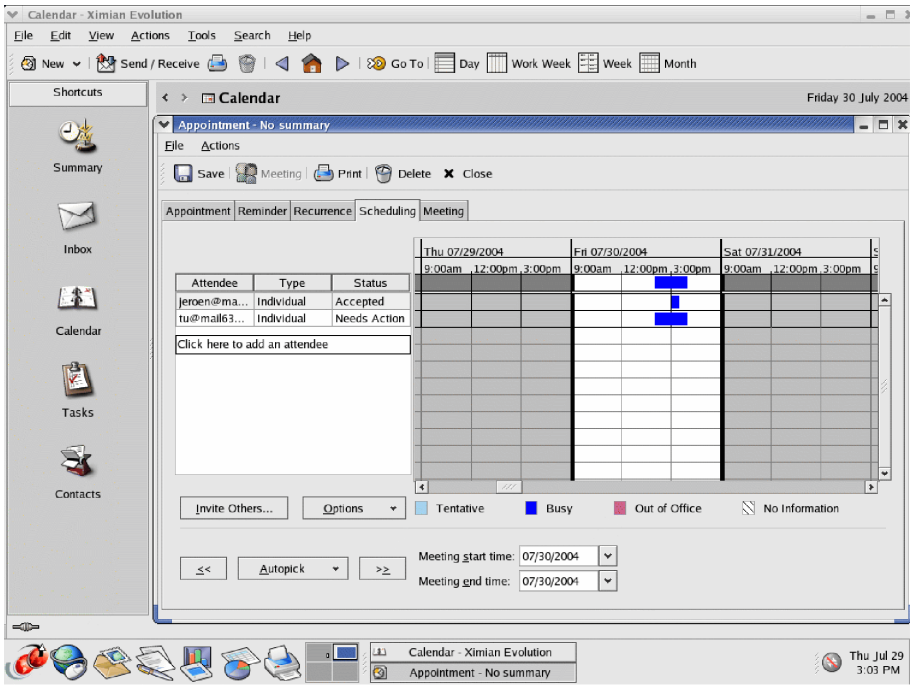

Рис. 10.3: Novell Evolution, установлено соединение с сервером Microsoft Exchange 2000

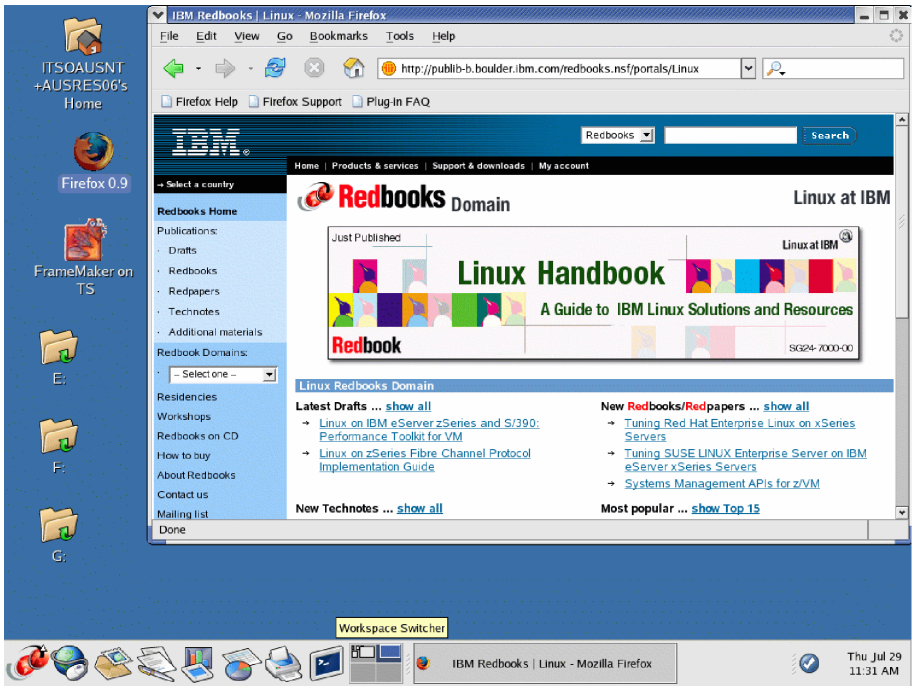

Рис. 10.4: Mozilla Firefox на Linux-клиенте

Рис. 10.5: Новый вид рабочего стола GNOME с произведенными для группы ITSO настройками

Глава 11

Рекомендации по интеграции

В этой главе описываются дополнительные методы интеграции Linux клиентов в существующие Windows домены (NT4 или Active Directory). Излагаются различные механизмы, включая подключение домашних каталогов для SMB ресурсов и аутентификация. Существуют два различных пути аутентификации пользователей в Windows домене:

- Используя winbind, входящий в Samba
- Используя прямое LDAP-соединение с Active Directory (это потребует изменений в схеме Active Directory)

Разделы этой главы:

- Разд. 11.1 Подключение к домену Windows
- Разд. 11.2 Как использовать winbind для подключения локальных пользователей
- Разд. 11.3 Как использовать LDAP для подключения к Active Directory
- Разд. 11.4 Pluggable Authentication Modules и домен
- Разд. 11.5 Как подключать файловые ресурсы к Linux-клиенту
- Разд. 11.6 Автоматическое монтирование домашних каталогов при входе в систему
- Разд. 11.7 Использование сетевых принтеров домена

Все примеры и рекомендации, приведенные в данной главе, соответствуют Samba версии не менее 3.0. Некоторая функциональность в более ранних версиях или отсутствует вовсе, или еще несовершенна.

11.1 Подключение к домену Windows

Многие how-to детально излагают рекомендации по добавлению к домену Linux серверов. Это how-to описывает технологию подключения к домену Linux-клиента. Добавление клиента к домену NT4 отличается от добавления клиента к Active Directory. Далее мы рассматриваем оба случая.

В большинстве примеров, приведенных в данном разделе, мы используем домен AD6380 как Primary Domain Controller SMB3LAB26 и Backup Domain Controller SMB3LAB27. В некоторых примерах (с использованием Windows 2003 Active Directory) мы используем домен AD6380 с контроллером домена W2K3AD.

11.1.1 Подключение к домену NT4

Для подключения к домену мы будем использовать Samba. Минимальный `smb.conf` приведен в Прим. 11.1.1.

Замените приведенные в примере домен и пароль на имя домена, используемое в вашей сети, и подправьте имена и адреса primary (и backup) домен-контроллеров.

Подключиться к домену вы можете следующим образом:

Пример 11.1.1: smb.conf для подсоединения к домену NT4

```
[global]
   workgroup = AD6380
   security = domain
   password server = SMB3LAB26,SMB3LAB27
```

```
net join -S SMB3LAB26 -U administrator
```

Замените SMB3LAB26 на имя (или IP) вашего primary домен-контроллера и используйте любую учетную запись домена, имеющую права на добавление машины к домену. Команда запросит пароль пользователя «administrator» данного домена.

Другие детали подключения к домену NT4 можно найти в Samba-HOWTO-Collection Samba-3. Подборку можно найти по адресу:

http://samba.org/samba/docs

11.1.2 Подключение к домену Active Directory

В этом случае необходимо использование Samba совместно с Kerberos. Kerberos нам понадобится для проведения аутентификации в Windows 200x KDC.

В примере мы будем использовать домен AD6380.LOCAL на AD сервере SMB3LAB26.

Необходимые для добавления в `smb.conf` строки приведены в Прим. 11.1.2.

Пример 11.1.2: smb.conf для подсоединения к домену Active Directory

```
[global]
   realm = AD6380.LOCAL
   workgroup = AD6380
   security = ads
   password server = SMB3LAB26
```

Убедитесь в том, что используете правильный регистр букв для realm, поскольку Kerberos к регистру чувствителен. Используйте полностью специфицированное имя (напр. AD6380.LOCAL) для realm и короткое имя для workgroup, в противном случае могут быть выданы предупреждения.

Минимальный krb5.conf может выглядеть как в Прим. 11.1.3.

Пример 11.1.3: krb5.conf для подсоединения к Windows 200x Kerberos realm

```
[libdefaults]
   default_realm = AD6380.LOCAL

[realms]
   AD6380.LOCAL = {
      kdc = SMB3LAB26:88
      admin_server = SMB3LAB26
   }

[domain_realm]
   .kerberos.server = AD6380.LOCAL
```

Убедитесь в том, что Kerberos сервер прописан в обратной зоне DNS как NetBIOS-имя KDC или NetBIOS-имя realm. Имя домена не должно подсоединяться к имени хоста. Простейший способ — внести соответствующую запись в `/etc/hosts`.

Поскольку Kerberos tickets крайне чувствительны к временным меткам, очень важно засинхронизировать AD-сервер и клиентов. Как Windows-клиенты получают время от домен-контроллера, так и Linux-клиент может использовать средства, предоставляемые Samba для синхронизации времени. Для этого можно воспользоваться командой **net time set**. Она запрашивает время у AD-сервера и соответственно устанавливает местные часы.

ВНИМАНИЕ

Убедитесь в том, что вы используете правильную версию Kerberos (MIT или Heimdal). Для AD на Windows Server® 2003 версия Heimdal должна быть не ниже 0.6, а MIT — 1.3.1. Убедитесь также в корректности команд kinit и klist и в том, что они не из поставки Java. Java kinit не работает.

В Novell Linux Desktop пакеты heimdal и heimdal-tools по умолчанию могут быть не установлены, но для подсоединения к AD в Windows 2003 они необходимы.

ВАЖНО

Убедитесь в том, что клиенты и сервер Active Directory (или Kerberos) имеют одно и тоже время, в рамках разрешенного разброса (skew).

Серверы Microsoft Windows используют Simple Network Time Protocol (SNTP), который является упрощенной версией (UNIX/Linux) Network Time Protocol (NTP). Форматы пакетов у них одинаковы и могут использоваться совместно. Это означает, что Linux-клиенты могут синхронизироваться с AD-серверами через ntp.

Конфигурацию Kerberos можно протестировать через **kinit USERNAME@REALM**, для того чтобы убедиться, что пароль принимается Windows 200x KDC.

ВНИМАНИЕ

В ADS будут работать только созданные вновь учетные записи или те записи, пароли которых были изменены в ходе миграции. Если записи в ходе миграции не менялись, команда kinit вернет сообщение о некорректности кода. Изменение пароля решает проблему.

Для того, чтобы действительно подсоединиться к домену AD, необходимо выполнить следующую команду:

```
net ads join -U administrator
```

Будет запрошен пароль администратора, например, для подключения клиентской машины client1 к домену AD6380 используется пользователь idsadmin, имеющий права администратора (см. Прим. 11.1.4).

Для подсоединения к необходимой организационной единице в первую очередь понадобится правильная комбинация имя-пароль. К примеру, если вы хотите подключиться к домену (то есть внести запись о компьютере) в контейнер Clients в организационном каталоге `Computers/ITSO`, вы должны выполнить:

```
kinit Administrator@AD6380.LOCAL
```

Пример 11.1.4: Пример подсоединения машины client1 к домену AD6380

```
[root@client1 root]# net ads join -U idsadmin
idsadmin password:*******
Using short domain name -- AD6380
Joined 'CLIENT1' to realm 'AD6380.LOCAL'
```

```
net ads join "Computers\ITSO\Clients"
```

ВНИМАНИЕ

Поскольку Windows 2003 использует подписи SMB, при подключении Windows 2003 ADS вы можете добавить в файл `smb.conf` следующую строчку:

```
client use spnego =
```

Для проверки подключения к AD-домену (во всех случаях) используйте команду:

```
net ads testjoin
```

Дополнительные детали о подключении к домену Active Directory можно найти в разделе 6.4 Samba HOWTO collection:

http://samba.org/samba/docs/Samba-HOWTO-Collection.pdf

11.2 Как использовать winbind для подключения локальных пользователей

После подключения к домену по процедуре, изложенной в Разд. 11.1, можно перенести все учетные записи домена на клиентское рабочее место для осуществления возможности авторизации для всех пользователей в домене. В небольших доменах это не проблема, но хорошей практикой администрирования это не является. Это означает, что добавление записи в домен должно тут же означать добавление такой же записи на всех клиентах. Это лишает всех преимуществ само использование домена.

В этом разделе мы опишем, как использовать winbind для избежания необходимости такого локального хранения. Демон winbind может транслировать учетные записи домена в uid'ы и gid'ы клиентской операционной системы.

11.2.1 Стандартный winbind

Демон winbind пользуется установками прописанными в файле `smb.conf`. В Прим. 11.2.1 приведены строки, которые надо добавить в `smb.conf`, для подключения winbind.

Данный разделитель (separator) означает, что учетная запись для пользователя Administrator в домене AD6380 будет записываться в виде: «AD6380+Administrator». Остальные записи задают диапазоны uid и gid для учетных записей домена, какие домашние каталоги и интерпретаторы будут использоваться при входе пользователя в систему. Как показывает практика, знак плюса (+) в качестве winbind-разделителя в Linux-окружении наиболее удобен.

Как альтернатива, разделитель и имя домена могут и вовсе не использоваться. Это означает, что используется домен по умолчанию. Для этого добавьте в `smb.conf` следующую строку:

```
winbind use default domain = yes
```

и возможно заменить template homedir на:

Пример 11.2.1: Добавления в smb.conf для winbind

```
[global]
   winbind separator = +
   idmap uid = 10000-20000
   idmap gid = 10000-20000
   winbind enum users = yes
   winbind enum groups = yes
   template homedir = /home/%D+%U
   template shell = /bin/bash
```

```
template homedir = /home/%U
```

Идентификаторы всех пользователей будут отображаться без имени домена и разделителя. Это может породить проблемы для локальных пользователей и пользователей домена, имеющих такой же идентификатор.

ПОДСКАЗКА

Установки имен с разделителем и именем домена могут породить проблемы при использовании приложений. Использование имени домена по умолчанию очень помогает при использовании pam_mount.

Записи idmap uid и gid указывают winbind в каких диапазонах следует порождать пользовательские и групповые идентификаторы для учетных записей домена. Выбирайте диапазоны, не пресекающиеся с локальными пользователями.

ВАЖНО

Если домен очень велик, у вас может не получиться разместить всех пользователей в указанном диапазоне. Эта проблема будет иметь место только при использовании на клиенте демона winbind, так как все отображение пространства пользователей идет через него.

После внесения указанных изменений, надо запустить winbind-демон. Обычно это выполняется командой:

```
/etc/init.d/winbind start
```

Убедитесь в том, что демон стартует при загрузке системы. В большинстве дистрибутивов (в первую очередь в Red Hat Enterprise Linux и Novell Linux Desktop) это достигается выполнением команды:

```
chkconfig winbind on
```

Работоспособность winbind может быть проверена командой **wbinfo**. Запустив ее с опцией -u, вы увидите все учетные записи домена, а с опцией -g — все группы.

ВАЖНО

Сервис nscd (name service caching daemon) конфликтует с winbind. Никогда не запускайте nscd на системе, где функционирует winbind.

Пример 11.2.2: Пример вывода команды wbinfo

```
[root@client1 root]# wbinfo -u
AD6380+Administrator
AD6380+Guest
AD6380+TsInternetUser
AD6380+idsadmin
AD6380+krbtgt
AD6380+HOST/client1
AD6380+SMB3LAB26$
```

С такими установками winbind сможет транслировать доменные учетные записи в локальных пользователей. Для того, что бы сообщить системе о необходимости использования winbind, следует внести соответствующие указания в конфигурацию Name Service Switch (NSS).

Конфигурационным файлом для NSS является `/etc/nsswitch.conf`. Он содержит записи вида:

```
passwd: files example
```

Это означает, что при проверке пароля запрос отправляется вначале на `/lib/libnss_files.so`, где производится поиск, а затем на `/lib/libnss_example.so`. Как только соответствующая запись будет найдена, результат возвращается приложению. Для того, что бы добавить поиск через winbind, необходимо добавить в `/etc/nsswitch.conf` следующие строки:

```
passwd: files winbind group: files winbind
```

Для проверки введите следующую команду:

```
getent passwd
```

Она возвратит не только записи из файла `/etc/passwd`, но и те, что были оттранслированы winbind.

Пример 11.2.3: Часть вывода getent passwd с подключенным в NSS winbind

```
[root@client1 root]# getent passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
.......................
AD6380+Administrator:x:10000:10000:Administrator:/home/AD6380+Administrator
:/bin/bash
AD6380+Guest:x:10001:10000:Guest:/home/AD6380+Guest:/bin/bash
AD6380+TsInternetUser:x:10002:10000:TsInternetUser:/home/AD6380+TsInternetU
ser:/bin/bash
AD6380+idsadmin:x:10003:10000:idsadmin:/home/AD6380+idsadmin:/bin/bash
AD6380+krbtgt:x:10004:10000:krbtgt:/home/AD6380+krbtgt:/bin/bash
AD6380+HOST/client1:x:10005:10001:client1:/home/AD6380+HOST/client1:/bin/ba
sh
AD6380+SMB3LAB26$:x:10006:10002:SMB3LAB26:/home/AD6380+SMB3LAB26_:/bin/bash
```

В силу алгоритма работы winbind (uid'ы присваиваются в том порядке, в котором пользователи авторизуются в домене, и это присвоение сохраняется в дальнейшем) отображение учетных записей домена в идентификаторы пользователей может быть различным на разных клиентских машинах. Это может привести к проблемам, в том случае, если клиенты используют разделяемые файловые системы на NFS. Механизмы, обсуждаемые в Разд. 11.2.2, позволяют избежать этих проблем.

ВАЖНО

В том случае, если исполнение winbind аварийно завершилось вследствие потери Kerberos-соединения из-за превышения time skew, демон перезапустится после его обновления.

Демон winbind занимается отображением пользователей, привязкой домашних каталогов и интерпретаторов. Сами домашние каталоги winbind не создаются. Как добиться такого создания, описано в Разд. 11.4.3.

Winbind используется для установления соединения с доменом и трансляции пользователей. Для того, чтобы дать возможность этим пользователям входить на Linux-клиента, нам понадобится изменить конфигурацию PAM, что подробно рассматривается в Разд. 11.4.1.

11.2.2 Альтернативы winbind

Главное отличие предлагаемых далее решений от изложенных выше, состоит в механизме генерации uid и gid. В предыдущем разделе winbind выбирал uid и gid для каждого пользователя из указанного диапазона. Очевидное следствие — их разный порядок на разных клиентах домена.

В этом разделе описываются два различных пути для решения этой проблемы:

- idmap_rid backend для генерации информации Active Directory
- idmap_ad backend для получения uid и gid из AD с расширенной схемой (Services for Unix)

Это относительно новые idmap backend'ы, и пока они являются экспериментальными. Но любой из них позволяет избавиться от недостатков, присущих winbind, при использовании в промышленных масштабах.

11.2.2.1 idmap_rid backend

Этот backend формирует uid для gid для пользователей домена, основываясь на доменных SID. Для включения этой опции, включите в `smb.conf` строки, приведенные в Прим. 11.2.4.

Пример 11.2.4: Пример smb.conf для использования idmap_rid backend

```
idmap uid = 10000-20000
idmap gid = 10000-20000
idmap backend = idmap_rid:AD6380=15000-20000
allow trusted domains = no
```

Используя эти установки и установив домен по умолчанию в «no» для полного отображения доменных учетных записей, мы можем наблюдать следующую картину:

Совершенно понятно, что вывод, приведенный в Прим. 11.2.5 отличается от того, что мы имели при выборке номеров для uid и gid из диапазона. Главное преимущество такой схемы отображения заключается в том, что это распределение одинаково для всех клиентов, подключенных к домену AD6380.

11.2.2.2 idmap_ad backend

В этом разделе излагается концепция LDAP, применяемая в Microsoft Services for Unix (SFU). Используемая в SFU схема для Active Directory расширена на uid и gid. Использование idmap_ad backend'а позволяет получать эту информацию и передавать ее winbind'у.

Это позволяет всем клиентам использовать одну и ту же информацию, поскольку отображением в данном случае занимается Active Directory.

Поддержка idmap_ad backend'а включается добавлением следующей строки в файл `smb.conf`:

```
idmap backend = idmap_ad
```

Пример 11.2.5: Частичный вывод getent passwd при использовании idmap_rid

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
.................................
AD6380+administrator:x:15500:15513:Administrator:/home/administrator:/bin/
bash
AD6380+guest:x:15501:15514:Guest:/home/guest:/bin/bash
AD6380+w2k3ad$:x:16003:15516:W2K3AD:/home/w2k3ad_:/bin/bash
AD6380+krbtgt:x:15502:15513:krbtgt:/home/krbtgt:/bin/bash
AD6380+testuser:x:16108:16121:Test User:/home/testuser:/bin/bash
AD6380+rhdt4$:x:16112:15515:rhdt4:/home/rhdt4_:/bin/bash
AD6380+rhdt4a$:x:16113:15515:rhdt4a:/home/rhdt4a_:/bin/bash
AD6380+ldapbind:x:16117:15513:ldapbind:/home/ldapbind:/bin/bash
AD6380+nld9$:x:16120:15515:nld9:/home/nld9_:/bin/bash
```

11.3 Как использовать LDAP для подключения к Active Directory

Active Directory реализует Lightwight Directory Access Protocol (LDAP). Вы можете использовать LDAP-клиент для подключения к AD для получения информации о пользователях. Однако, поскольку хранение данных о пользователях Unix/Linux в AD не предусматривалось, некоторые необходимые объекты в стандартной схеме отсутствуют. Одним из первых шагов по расширению схемы является добавление необходимых классов объектов.

Простейший путь — установка "Services for Unix"(SFU) от Microsoft. Данный продукт включает в себя все необходимые расширения для хранения таких данных.

Установка SFU с использованием параметров по умолчанию включает в себя следующее:

1. Распаковать файлы в каталог, иной чем `C:\SFU`, например, в `C:\Temp\SFU`
2. Запустить **setup.exe**
3. Выбрать Standard Installation
4. На экране Security Settings оставить обе опции пустыми
5. Выбрать «Local Username Mapping Server» и «Network Information Services»
6. Выбрать имя домена
7. Перезагрузить AD-сервер

В результате мы расширяем схему AD, включая в нее такие поля как uid и gid.

Следующим шагом мы создаем пользователя в Active Directory, используя LDAP-клиента для подключения к каталогу. Установите секретный пароль и пометьте его как «not expire».

Установленные к схеме дополнения позволяют теперь задавать gid у групп и uid, shell и домашний каталог для пользователей, см. Рис. 11.1.

В этом примере создан пользователь testuser с uid 5000, shell'ом и домашним каталогом. Также мы создали группу linuxusers с gid 5000.

Дополнительную информацию о Services for Unix и интеграции Active Directory с каталогами для Linux/UNIX-клиентов можно найти в документе *Solution Guide for Windows Security and Directory Services for UNIX* от Microsoft.

Для того чтобы система использовала LDAP-подключение к Active Directory, надо задать соответствующую конфигурацию в файле `/etc/ldap.conf`. Эта конфигурация должна включать в себя следующую информацию:

- IP адрес сервера Active Directory (разделяются пробелами, если их несколько)

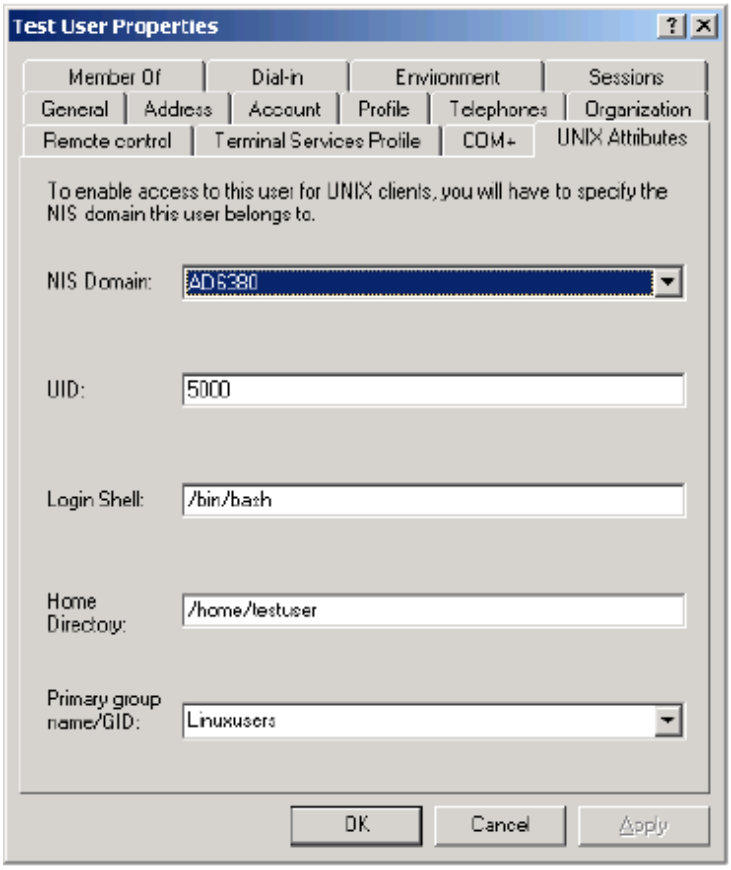

Рис. 11.1: Окно Active Directory User Properties с Unix Attributes

- Базовый узел (bind node) в дереве каталога для начала поиска
- Пользователь (bind user), через которого осуществляется подсоединение к AD
- Пароль этого пользователя (открытым текстом, так что берегите свой конфигурационный файл)
- Границы поиска
- SSL подключение
- Записи, описывающие отображение объектов доменной схемы, необходимых для NSS

Пример конфигурационного файла см. на Прим. 11.3.1.

Следующие записи в NSS-конфигурации изменены для включения возможности входа в систему через LDAP:

```
passwd: files ldap group: files ldap
```

Для проверки запустите команду:

```
getent passwd
```

Теперь покажем, как выглядит запись пользователя testuser при выводе passwd, см. Прим. 11.3.2.

ВНИМАНИЕ

Заметьте, что пользователи отображаются без имени домена и символа-разделителя. Это так же означает, что перекрытие имен исключено, так как пользователь один и тот же и в локальных файлах, и в домене.

Пример 11.3.1: Фрагмент файла /etc/ldap.conf

```
host 192.168.100.110
base cn=Users,dc=AD6380,dc=LOCAL
binddn cn=ldapbind,cn=Users, dc=AD6380,dc=LOCAL
bindpw Welcome2006
scope sub
ssl no
nss_base_passwd cn=Users,dc=AD6380,dc=LOCAL?sub
nss_base_shadow cn=Users,dc=AD6380,dc=LOCAL?sub
nss_base_group cn=Users,dc=AD6380,dc=LOCAL?sub
nss_map_objectclass posixAccount user
nss_map_objectclass shadowAccount user
nss_map_attribute uid sAMAccountName
nss_map_attribute uidNumber msSFU30UidNumber
nss_map_attribute gidNumber msSFU30GidNumber
nss_map_attribute loginShell msSFU30LoginShell
nss_map_attribute gecos name
nss_map_attribute userPassword msSFU30Password
nss_map_attribute homeDirectory msSFU30HomeDirectory
nss_map_objectclass posixGroup Group
nss_map_attribute uniqueMember msSFU30PosixMember
nss_map_attribute cn cn
```

Пример 11.3.2: Фрагмент вывода getent passwd при включенном ldap в NSS

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
.............................
testuser:ABCD!efgh12345$67890:5000:5000:Test User:/home/testuser:/bin/bash
```

Таким образом, вместо генерации записей с uid и gid на каждом клиенте мы при помощи такого метода получаем эти записи из Active Directory. Однако это означает, что эти записи должны быть заведены именно там с самого сначала.

Различия, достоинства и недостатки обоих методов, winbind и LDAP, мы уже обсуждали в Разд. 7.2.

Часть информации данного раздела базируется на:

http://enterprise.linux.com/enterprise/04/12/09/2318244.shtml?tid=102&tid=101&tid=100

11.4 Pluggable Authentication Modules и домен

Подсоединив Linux-клиент к домену (согласно Разд. 11.1) и зная, как воспользоваться доменными учетными записями с использованием winbind (Разд. 11.2) или LDAP (Разд. 11.3), вы можете сконфигурировать систему для входа через доменные записи.

Pluggable Authentication Modules (PAM) представляет определенный уровень абстракции для процедур аутентификации и авторизации. Используя PAM-модули можно менять методы аутентификации и авторизации без перекомпиляции приложений.

В большинстве дистрибутивов конфигурация PAM задается в файлах `/etc/pam.d/`. Таким образом, изменение режима аутентификации достигается простым добавлением соответствующего модуля в конфигурационный файл.

11.4.1 Аутентификация пользователей с использованием winbind и PAM

Поскольку реализации PAM в различных дистрибутивах Linux отличаются друг от друга, попробуем обсудить детали тестируемых дистрибутивов.

11.4.1.1 Winbind и PAM в Red Hat Desktop

В Red Hat реализован модуль pam_stack. Это означает, что конфигурационные файлы из `/etc/pam.d/` могут использовать установки из других файлов, включая вложенные конфигурации. Для этих целей в Red Hat используются установки в конфигурационном файле `/etc/pam.d/system-auth`. Это означает, что едва мы только добавим PAM модули winbind в этот файл, как все приложения использующие PAM получат winbind поддержку.

Поскольку файл `/etc/pam.d/system-auth` генерируется автоматически через **authconfig**, будьте осторожны при запуске этой команды после того, как внесете в него изменения вручную.

Пример файла `system-auth`, содержащего необходимые строки, приведен в Прим. 11.4.1.

Пример 11.4.1: Фрагмент файла /etc/pam.d/system-auth с добавлениями для winbind-аутентификации

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth    required   /lib/security/$ISA/pam_env.so
auth    sufficient /lib/security/$ISA/pam_unix.so likeauth nullok

auth    sufficient /lib/security/$ISA/pam_winbind.so use_first_pass
auth    sufficient /lib/security/$ISA/pam_krb5.so use_first_pass
auth    sufficient /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth    required   /lib/security/$ISA/pam_deny.so
account required   /lib/security/$ISA/pam_unix.so

account sufficient /lib/security/$ISA/pam_winbind.so
.......................
```

Поскольку pam_winbind.so был включен в файл `system-auth`, все приложения, использующие этот файл через pam_stack.so, получают поддержку winbind. Это означает, что вы сможете входить на Linux-клиента, используя доменные учетные записи следующей нотации:

```
<domainname><winbind-seperator><accountname>
```

Так, например, в нашем тестовом домене это будет выглядеть как AD6380+Administrator.

Если при использовании каких-то приложений возникают проблемы после подключения winbind через модуль pam_stack, то вызов pam_winbind.so может быть помещен в конфигурационные файлы только тех приложений, для которых это необходимо.

11.4.1.2 Winbind и PAM в Novell Linux Desktop

В Novell Linux Desktop (NLD) не используется модуль pam_stack для получения установок из центрального файла. Это означает, что для каждого приложения понадобится добавление модуля winbind аутентификации в его конфигурационный файл в `/etc/pam.d/`.

В качестве примера мы приводим конфигурационный файл sshd. После включения модуля pam_winbind.so он будет выглядеть как в Прим. 11.4.2

Для каждого поддерживающего PAM приложения, которое вы хотите сделать доступным для пользователей домена, необходимо добавить строки auth и account для pam_winbind.so (до строк nologin). Для предотвращения

Пример 11.4.2: Конфигурационный файл /etc/pam.d/sshd после подключения winbind

```
#%PAM-1.0
auth       sufficient pam_winbind.so
auth       required   pam_unix2.so use_first_pass # set_secrpc
auth       required   pam_nologin.so
auth       required   pam_env.so
account    required   pam_unix2.so
account    sufficient pam_winbind.so
account    required   pam_nologin.so
password   required   pam_pwcheck.so
password   required   pam_unix2.souse_first_pass use_authtok
session    required   pam_unix2.sonone # trace or debug
session    required   pam_limits.so
```

двойного запроса пароля ко всем модулям, запрашивающим пароль, в конфигурационный файл должен быть добавлен параметр use_first_pass кроме модуля pam_winbind.so.

11.4.2 Как производить аутентификацию пользователей через LDAP и PAM

Для того чтобы подсистема PAM могла проводить аутентификацию пользователей через LDAP, для LDAP надо произвести дополнительное конфигурирование.

Поскольку объекты SFU-расширения схемы Active Directory имеют иные названия, чем требуется для PAM, нам понадобится настроить отображение. См. Прим. 11.4.3

Пример 11.4.3: Отображение для файла /etc/ldap.conf

```
pam_login_attribute sAMAccountName
pam_filter objectclass=user
pam_member_attribute msSFU30PosixMember
pam_groupdn cn=unixusergroup,dc=AD6380,dc=LOCAL
pam_password ad
```

Выделенные в Прим. 11.4.3 строки ограничивают набор пользователей AD, которым разрешен вход на Linux-клиент. PAM разрешит вход только членам группы unixusergroup в AD.

Поскольку детали реализации механизма PAM различны в разных дистрибутивах Linux, рассмотрим эти детали поподробнее.

11.4.2.1 LDAP и PAM в Red Hat Desktop

В Red Hat используется модуль pam_stack. Это означает, что конфигурационные файлы в `/etc/pam.d/` могут использовать установки из других файлов. По этой причине, в Red Hat многие установки делаются в конфигурационном файле `/etc/pam.d/system-auth`. Это означает, что нам достаточно добавить LDAP PAM модули в этот файл и все приложения, использующие PAM, тотчас получают доступ к LDAP. PAM-модуль носит название pam_ldap.

Поскольку `/etc/pam.d/system-auth` генерируется **authconfig**, будьте внимательны, запуская эту команду после того, как изменили этот файл вручную.

Пример файла `system-auth`, содержащего эти строки, см. в Прим. 11.4.4.

После включения pam_ldap.so в system-auth все приложения, использующие этот файл через pam_stack.so, становятся способными работать с LDAP. Это означает, что мы можем входить на Linux-клиента через учетные записи домена.

Пример 11.4.4: Часть файла /etc/pam.d/system-auth с включением LDAP аутентификации

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/$ISA/pam_env.so
auth        sufficient    /lib/security/$ISA/pam_unix.so likeauth nullok
auth        sufficient    /lib/security/$ISA/pam_ldap.so use_first_pass
auth        sufficient    /lib/security/$ISA/pam_krb5.so use_first_pass
auth        required      /lib/security/$ISA/pam_deny.so
account     required      /lib/security/$ISA/pam_unix.so
account     required      /lib/security/$ISA/pam_ldap.so
.........................
```

Если подключение пользователей домена ко всем приложениям через модуль pam_stack вызывает проблемы, вызов pam_ldap.so должен быть помещен только в конфигурационные файлы тех приложений, для которых это необходимо.

ВНИМАНИЕ

Убедитесь, что вы поместили pam_ldap, как это требуется в раздел account, иначе даже пользователь, не входящий в группу, указанную в /etc/ldap.conf, сможет осуществить вход.

В Прим. 11.4.5 показан вывод команды ssh при входе пользователя из домена, который не входит в группу unixusergroup (как описано в Прим. 11.4.4), но поля account для pam_ldap оказывается достаточно.

Пример 11.4.5: Успешный вход через ssh пользователя, не входящего в правильную группу

```
[root@RHDT4a ~]# ssh testuser@localhost
testuser@localhost's password:
You must be a msSFU30PosixMember of cn=unixusergroup,dc=AD6380,dc=LOCAL to
login.
Last login: Fri Jan 13 10:09:34 2006 from rhdt4a
-bash-3.00$
```

В Прим. 11.4.6 показан вывод той же команды в случае, когда поле account для pam_ldap востребовано. Вход в систему не происходит, и нам выдается соответствующее сообщение.

Пример 11.4.6: Неуспешный вход через ssh, вследствие неправильного членства в группе

```
[root@RHDT4a ~]# ssh testuser@localhost
testuser@localhost's password:
Permission denied, please try again.
```

11.4.2.2 LDAP и PAM в Novell Linux Desktop

Novell Linux Desktop (NLD) не использует модуль pam_stack для получения установок из центрального файла. Это значит, что для каждого приложения, которому понадобится использовать модуль LDAP-аутентификации, потребуется добавить его в соответствующий конфигурационный файл в `/etc/pam.d`.

В качестве примера мы возьмем конфигурационный файл sshd. После добавления в него модуля pam_ldap.so он выглядит как в Прим. 11.4.7.

Пример 11.4.7: Конфигурационный файл /etc/pam.d/sshd после добавления LDAP

```
#%PAM-1.0
auth       sufficient pam_ldap.so
auth       required   pam_unix2.so use_first_pass # set_secrpc
auth       required   pam_nologin.so
auth       required   pam_env.so
account    required   pam_unix2.so
account    required   pam_ldap.so
account    required   pam_nologin.so
password   required   pam_pwcheck.so
password   required   pam_unix2.souse_first_pass use_authtok
session    required   pam_unix2.sonone # trace or debug
session    required   pam_limits.so
```

Для каждого приложения, поддерживающего PAM, для которого вы хотите открыть доступ пользователям домена, необходимо добавить строки auth и account для pam_ldap.so. Для того чтобы не надо было вводить пароль дважды, в конфигурационный файл для каждого PAM-модуля должен быть добавлен параметр use_first_pass, за исключением модуля pam_ldap.so.

> ВНИМАНИЕ
>
> Убедитесь, что вы поместили pam_ldap в разделе account, иначе группа, указанная в /etc/ldap.conf, не будет действовать.

11.4.3 PAM и домашние каталоги

Пользователи, входящие в домен на клиенте, локально не существуют. Через SFU-расширение схемы AD или winbind конфигурацию `/etc/samba/smb.conf` мы сообщаем системе, какую командную оболочку следует использовать и где находится домашний каталог пользователя (для локальных пользователей эта информация располагается в файле `/etc/passwd`). Однако, домашний каталог ни winbind, ни LDAP не создает.

Эта проблема может быть решена разными путями:

- Создать все возможные домашние каталоги (пустые) на всех клиентах.
- Создать все домашние каталоги на файловой системе сервера, которая монтируется всеми клиентами (по SMB или NFS).
- Использовать модуль pam_mkhomedir.so, который создает домашний каталог при первым входе пользователя в систему.

Первые два варианта кажутся наиболее очевидными, но усложняют дальнейшее обслуживание системы при добавлении пользователя к домену. Последний вариант является наилучшим выходом и состоит в добавлении в конфигурационные файлы PAM строки, создающей домашний каталог в случае его отсутствия. Это изменение вносится в конфигурационный файл PAM однажды, когда создается Linux-клиент «мастер».

11.4.3.1 Red Hat Desktop

В системах Red Hat модуль pam_mkhomedir добавляется в `/etc/pam.d/system-auth`. После этого домашний каталог будет создаваться при входе пользователя в систему, например, через терминальный вход, secure shell session или графическую сессию.

Строки, которые надо добавить в этот файл, представлены в Прим. 11.4.8.

Пример 11.4.8: Фрагмент файла /etc/pam.d/system-auth, включающий pam_mkhomedir

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth    required   /lib/security/$ISA/pam_env.so
auth    sufficient /lib/security/$ISA/pam_unix.so likeauth nullok
auth    sufficient /lib/security/$ISA/pam_winbind.so use_first_pass
auth    sufficient /lib/security/$ISA/pam_krb5.so use_first_pass
auth    sufficient /lib/security/$ISA/pam_smb_auth.so use_first_pass nolocal
auth    required   /lib/security/$ISA/pam_deny.so
account required   /lib/security/$ISA/pam_unix.so
account sufficient /lib/security/$ISA/pam_winbind.so
.........................

session optional  /lib/security/$ISA/pam_mkhomedir skel=/etc/skel umask=0022
```

Опция skel сообщает модулю, где брать skeleton-файлы для копирования их во вновь созданные домашние каталоги. Опция umask указывает на установки прав при создании каталога и его содержимого.

11.4.3.2 Novell Linux Desktop

Поскольку NLD не использует system-auth и pam_stack.so, pam_mkhomedir.so следует добавить в каждый конфигурационный файл приложений, которые осуществляют вход пользователя в систему. Примерами таких приложений могут являться ssh, telnet, gdm, xdm и login. Первый вход каждого пользователя домена через любое из этих приложений будет приводить к созданию домашнего каталога.

Строка, которую следует добавить в файл, выглядит следующим образом:

```
session optional pam_mkhomedir skel=/etc/skell umask=0022
```

Пример для ssh см. в Прим. 11.4.9.

Опция skel сообщает модулю, где брать skeleton-файлы для копирования их во вновь созданные домашние каталоги. Опция umask указывает на установки прав при создании каталога и его содержимого.

В Прим. 11.4.10 показано, как создается домашний каталог, когда доменный пользователь testuser впервые входит в систему.

11.5 Как подключать файловые ресурсы к Linux-клиенту

Этот раздел описывает различные пути подключения Windows-ресурсов.

Один вариант заключается в постоянном подсоединении к ресурсу путем монтирования куда-то в файловую систему. Эти ресурсы могут подключаться через smb mount CIFS mount'a.

Если монтирование невозможно, можно воспользоваться командой smbclient. Smbclient работает как ftp-клиент и может использоваться для получения и загрузки файлов на Windows-ресурсах.

Пример 11.4.9: Пример файла /etc/pam.d/sshd с включенным pam_mkhomedir.so

```
#%PAM-1.0
auth       sufficient pam_ldap.so
auth       required   pam_unix2.so use_first_pass
auth       required   pam_nologin.so
auth       required   pam_env.so
account    required   pam_unix2.so
account    required   pam_ldap.so
account    required   pam_nologin.so
password   required   pam_pwcheck.so
password   required   pam_unix2.so use_first_pass use_authtok
session    optional   pam_mkhomedir.so skel=/etc/skel umask=0022
session    required   pam_unix2.so none         # trace or debug
session    required   pam_limits.so
```

Пример 11.4.10: Ввод при первом входе пользователя домена в систему

```
nld9:~ # ssh testuser@localhost
Password:
Creating directory '/home/testuser'.
Creating directory '/home/testuser/bin'.
Creating directory '/home/testuser/Documents'.
Creating directory '/home/testuser/public_html'.
Creating directory '/home/testuser/.xemacs'.
Creating directory '/home/testuser/.fonts'.
Last login: Fri Jan 13 16:09:47 2006 from localhost
testuser@nld9:~> pwd
/home/testuser
```

11.5.1 Подключение ресурсов через smbfs

Ресурс может быть подключен при помощи команды **mount** следующим образом:

```
mount -t smbfs //servername/sharename /mountpoint -o user=domainuser
```

Команда **mount** вызывает **smbmount** для монтирования ресурса `sharename` на сервере `servername` в точку монтирования `mountpoint`. При подключении используется пользователь `domainuser` и в ходе исполнения команды запрашивается его пароль.

Это наиболее распространенный способ подключения Windows-ресурсов к не Windows операционным системам при помощи Samba.

Однако, если ресурс помещен на сервере, который выполняет SMB signing (а многие Windows делают это), исполнение команды прервется, выдав сообщение об ошибке. Это следствие того, что Samba-клиент не поддерживает SMB signing и не может подмонтировать ресурс.

11.5.2 Подключение ресурсов через CIFS

Common Internet File System (CIFS) это платформо-независимая система разделения файловых ресурсов, построенная по принципам SMB файловых систем Microsoft. В новых версиях 2.6 ядра Linux CIFS встроен. Подключение ресурсов как файловых систем CIFS позволяет избежать описанных выше проблем с SMB signing для Linux-клиентов.

Кроме того, CIFS mount может использоваться для подключения файловых систем Microsoft DFS™, smbmount для которых не работает.

Синтаксис использования mount для CIFS очень похож на монтирование при помощи smbfs:

```
mount -t cifs //servername/sharename /mountpoint -o user=domainuser
```

> **ПОДСКАЗКА**
>
> Ориентируясь на будущее, старайтесь использовать CIFS везде, где это возможно

11.5.3 Использование smbclient

Команда smbclient, по сути дела, превращает Windows-ресурс в ftp-сервер. Работает она следующим образом:

```
smbclient //servername/sharename -U domainuser -W domainname
```

Команда запрашивает пароль и соединяется с ресурсом. Вот некоторые из команд, которые можно использовать в smbclient:

- get — выкачивает файл, расположенный на ресурсе, в локальную файловую систему
- put — копирует локальный файл на ресурс
- cd — смена рабочего каталога на ресурсе
- lcd — смена рабочего каталога в локальной файловой системе
- help — показать все доступные команды
- quit — выход из smbclient

В Прим. 11.5.1 показано подключение к ресурсу и получение подсказки по использованию команды **smbclient**.

Пример 11.5.1: Пример использования smbclient для подключения к Windows-ресурсу

```
nld9:~ # smbclient //w2k3ad.ad6380.local/share1 -U administrator
Password:
Domain=[AD6380] OS=[Windows Server 2003 3790] Server=[Windows Server 2003
 5.2]
smb: \> help get
HELP get:
        <remote name> [local name] get a file
smb: \>
```

11.6 Автоматическое монтирование домашних каталогов при входе в систему

Одной из замечательных возможностей Windows является функция Single Sign On (SSO). При входе в ОС Windows у вас запрашивается пароль и в дальнейшем этот пароль используется при подключении ресурсов, помеченных как подсоединяемые при входе.

Подобной функциональности можно достичь и в случае Linux-клиента при помощи модуля pam_mount. Модуль этот пока находится в стадии разработки и включен не во все enterprise-дистрибутивы. Но немного потрудившись, можно заставить его работать.

ВНИМАНИЕ

С сентября 2005 pam_mount'у уделяется заметно больше внимания. Новые версии выходят чаще. Несмотря на это, еще не во всех дистрибутивах он работает.

Файловые системы SMB и CIFS не позволяют создавать символьные ссылки или сокеты. Это следствие того, что Windows-ресурсы, к которым они подсоединяются, таких возможностей не поддерживают. Это означает, что приложения, которым необходимо такие объекты создавать, не смогут работать на этих файловых системах.

ВНИМАНИЕ

Будьте внимательны при монтировании SMB-ресурсов в качестве домашних каталогов пользователей при использовании графического входа в систему. Некоторые графические окружения не могут корректно работать на файловых системах, подключенных через SMB, поскольку они очень зависимы от символьных ссылок и сокетов.

ПОДСКАЗКА

Монтирование пользовательского доменного ресурса в качестве подкаталога домашнего каталога позволяет избежать всех проблем с графическими окружениями.

Модуль pam_mount монтирует не только файловые системы SMB и CIFS, но еще и NCP, шифрованные loop файловые системы и практически любые файловые системы, поддерживаемые командой **mount**.

11.6.1 pam_mount на Red Hat Desktop

Модуль, добавляемый в `/etc/pam.d/system-auth`, включает автоматическое монтирование для всех вариантов входа в систему. Модуль включает в себя часть «auth», которая получает пароль через PAM; и часть «session», которая собственно и занимается монтированием. Также этот сессионный модуль занимается отключением ресурсов при выходе из сессии.

ПОДСКАЗКА

Убедитесь в том, что вам действительно необходимо автоматическое монтирование при всех видах входа в систему. Например, su от пользователя root на какого-то пользователя может не сработать, так как пароль при этом не запрашивается.

Строки, которые следует добавить:

```
auth required pam_mount.so
```

и:

```
session optional pam_mount.so
```

Строка auth должна предшествовать строкам pam_unix и pam_winbind. Строка session должна присутствовать в соответствующем разделе. Не забудьте добавить аргумент `use_first_auth` для строки, не имеющей перед собой pam_mount, иначе пароль будет запрашиваться дважды.

Модуль pam_mount имеет свой собственный конфигурационный файл `/etc/security/pam_mount.conf`. Этот файл содержит установки, касающиеся того, как выполнить команды mount и umount, установки режима отладки и указания какие файловые системы должны подключаться при входе тех или иных пользователей.

Минимальный pam_mount.conf только для монтирования SMB-ресурсов выглядит следующим образом, см. Прим. 11.6.1.

Пример 11.6.1: Минимальный pam_mount.conf

```
debug 1
mkmountpoint 1
options_require     nosuid,nodev
lsof /usr/sbin/lsof
fsck /sbin/fsck
losetup /sbin/losetup
unlosetup /sbin/losetup -d
smbmount /usr/bin/smbmount
umount   /usr/bin/smbumount
volume * smb smb3lab26 & /home/&/domainshare uid=&,dmask=0750 - -
```

Таким образом включается режим отладки для pam_mount. Точки монтирования создаются в том случае, если они не существуют, и ресурс подключается согласно описанию, представленному в последней строке. Звездочка в начале последней строки в примере, приведенном выше показывает, что данный ресурс монтируется любым пользователем. Амперсанд (&) в определении расширяется до имени пользователя. Так строка volume в Прим. 11.6.1 сообщает pam_mount'у о необходимости подключить сетевой ресурс, прописанный после имени пользователя, с сервера smb3lab26 в точку монтирования `/home/<username>/domainshare`. Опции, указанные в строке options говорят о том, что владельцем подмонтированного ресурса будет uid, и права доступа будут определяться dmask.

Внимание

В некоторых случаях модуль pam_mount не будет отрабатывать при попытке выполнить setuid root, что означает, что нормальные команды монтирования будут прерываться с сообщением, что только суперпользователь root может выполнять такие операции. Однако это явление не частое.

Возможная проблема при монтировании ресурсов домена заключается в том, что в файле `pam_mount.conf` амперсанд (&) расширяется в полное имя пользователя, включая имя домена и разделитель. Это великолепно работает с pam_mount в том случае, когда у winbind установлена опция «default domain = yes» в `smb.conf`, для получения имен пользователей домена без домена и разделителя. В ином случае понадобятся дополнительные скрипты для их удаления из имени пользователя перед выполнением монтирования.

ПОДСКАЗКА

При работе с pam_mount рекомендуется использование простых имен пользователей для доменных записей. Это означает, что следует сделать установки для default domain. См. Разд. 11.2.1

11.6.2 pam_mount в Novell Linux Desktop

Ситуация при использовании Novell Linux Desktop и pam_mount несколько отличается от Red Hat Desktop. На Novell Linux Desktop собраны последние версии pam_mount. Однако использование pam_mount с sshd гораздо сложнее.

Pam_mount добавляется во все файлы `/etc/pam.d` для всех сервисов, требующих автоматического монтирования. Модуль включает в себя как часть «auth», получающую пароль через PAM; так и часть «session», собственно и осуществляющую монтирование. Сессионный модуль так же занимается отключением ресурсов при выходе из сессии.

ВНИМАНИЕ

Поскольку некоторые версии OpenSSH разбивают аутентификацию на отдельные процессы, пароль не может быть передан из аутентификационной части pam_mount в сессионную. Это означает, что pam_mount в этих случаях не будет работать вовсе.

Строки, которые надо добавить, будут иметь вид:

```
auth required pam_mount.so
```

и:

```
session optional pam_mount.so
```

Строка auth должна предшествовать строкам pam_unix и pam_winbind. Строка session должна располагаться в соответствующем разделе. Помните, что необходимо добавление аргумента `use_first_auth` для строки, не имеющей перед собой pam_mount, иначе пароль будет запрашиваться дважды.

Модуль pam_mount имеет свой собственный конфигурационный файл `/etc/security/pam_mount.conf`. Этот файл содержит установки команд монтирования, режимы отладки, точки монтирования, которые надо создавать, и указания, какие файловые системы должны подключаться при входе тех или иных пользователей.

Минимальный pam_mount.conf только для монтирования SMB-ресурсов выглядит так же как и в Прим. 11.6.1.

11.7 Использование сетевых принтеров домена

Использование имеющихся принтеров является одним из главных предварительных условий включения новых типов клиентов в существующее окружение. Мы не хотим поддерживать различные окружения для разных типов клиентов.

Это означает, что Linux-клиенты должны быть способны использовать сетевые принтеры доступные в домене. Этого можно достичь следующими путями:

- Печать непосредственно на принтер, используя его сетевой интерфейс.
- Печать на принтере через Windows print server и его SMB-интерфейс.

Оба варианта доступны при использовании CUPS на Linux-клиенте. Мы рекомендуем использовать второй вариант, однако это возможно только в том случае, если все системы печати в домене (и пользовательские окружения) построены на одной базе. Это означает, что речь идет только о Windows и Linux системах печати.

На Linux-клиенте должен работать CUPS. Это можно проверить так:

```
/etc/init.d/cups status
```

В результате должен быть возвращен PID процесса CUPS. Если он не запущен, его можно запустить используя следующую команду:

```
/etc/init.d/cups start
```

Убедитесь в том, что он запустится после следующей перезагрузки:

```
chkconfig cups on
```

Вы также должны убедиться в том, что присутствует ссылка на smbspool в каталоге backend'ов CUPS:

```
# ls -l /usr/lib/cups/backend/smb
lrwxrwxrwx 1 root root 21 Mar 1 12:34 /usr/lib/cups/backend/smb ->
../../../bin/smbspool
```

Вы можете проверить, поддерживает ли CUPS smb, при помощи команды:

```
# lpinfo -v | grep smb
network smb
```

Вы можете создать принтер двумя различными способами:

- Используя команду lpadmin
- Через веб-интерфейс CUPS

Мы опишем оба варианта детальнее.

11.7.1 Создание принтера при помощи команды lpadmin

Принтер можно создать из командной строки. Это дает возможность создания большого количества принтеров командным скриптом. Это означает, что создание доменных принтеров может быть встроено в пользовательский профиль, так что пользователям не понадобится создавать принтер вручную на каждом клиенте.

Для создания принтера задайте следующую команду:

```
lpadmin -p printer1 -E -v smb://Administrator:*****@SMB3LAB26/printer1 \
    -D "printer1 in AD6380 example domain"
```

Параметры, конечно, надо будет заменить на имена в вашем домене. Первое вхождение printer1, это название локального принтера, а второе — название ресурса. Сервер SMB3LAB26 — это сервер печати. Если принтер доступен всем, можете оставить имя пользователя и пароль в smb: URL. Поскольку использование привязанных к пользователю ресурсов печати может привести к раскрытию паролей через скрипты или login-профайлы, лучшей практикой является открытие принтерных ресурсов для всех.

Кроме того, вы можете добавить опцию -m для указания модели, или опцию -l для указания места расположения.

11.7.2 Создание принтера через веб-интерфейс CUPS

Все то же самое можно проделать через веб-интерфейс CUPS. Для этого нам нужно выполнить следующие шаги:

1. В броузере загрузите административный интерфейс CUPS, используя URL:

   http://localhost:631/.

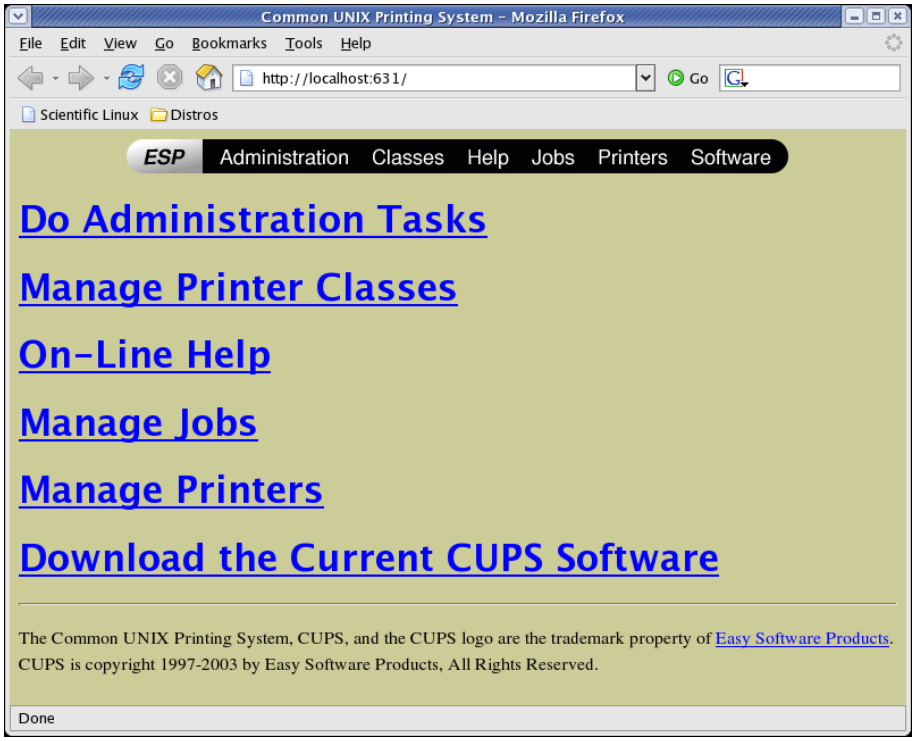

Рис. 11.2: Стартовая страница веб-интерфейса CUPS

2. Нажмите Manage Printers (Управление принтерами)
3. Нажмите Add (Добавить)

   Вы получите страницу, на которой вы сможете ввести Name (Имя), Location (Местоположение) и Description (Описание) принтера. Теперь нажмите Continue (Продолжить).

4. На следующей странице вы выберете устройство, к которому подключен принтер. Выберите «Windows Printer via SAMBA» и нажмите Continue (Продолжить).
5. На следующей странице у вас запросят URI устройства. Введите:

   ```
   smb://<domainname>:<password>@<printserver>/<printershare>
   ```

   Или :

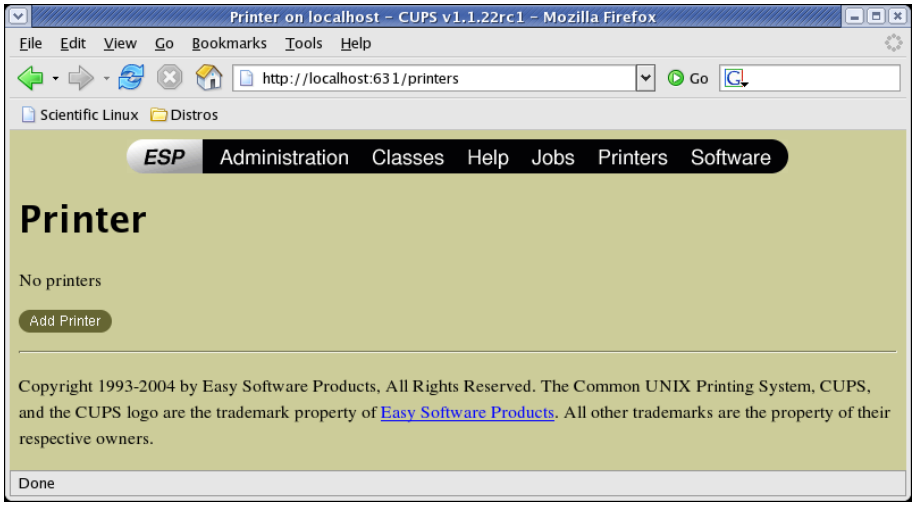

Рис. 11.3: Страница управления веб-интерфейса CUPS

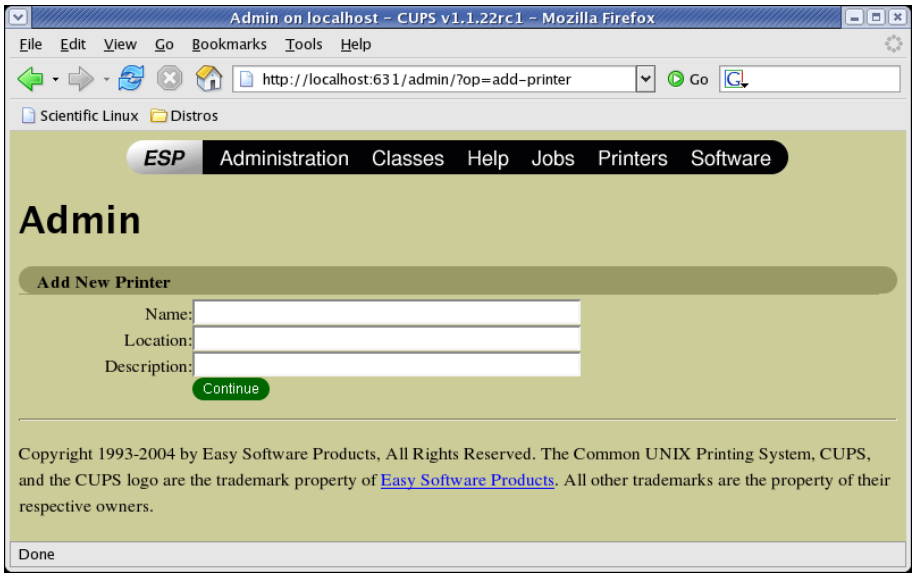

Рис. 11.4: Страница добавления принтера веб-интерфейса CUPS

`smb://<printserver>/<printershare>`

А затем нажмите Continue (Продолжить).

6. На следующей странице выберите модель и драйвер принтера и нажмите Continue (Продолжить).

   На следующей странице вы можете сделать более детальный выбор на основе модели, выбранной на предыдущем экране. Выберите нужный и нажмите Continue (Продолжить).

7. Вы добавили принтер в CUPS.

Скриншоты, показанные в этом разделе, созданы с использованием клиента Red Hat Desktop, подсоединенного к домену Active Directory AD6380, с домен-контроллером smb3lab26.

Для тестирования добавленного вами через **lpadmin** или через веб-интерфейс принтера просто напечатайте файл или воспользуйтесь кнопкой Print Test Page (Напечатать Тестовую Страницу) веб-интерфейса.

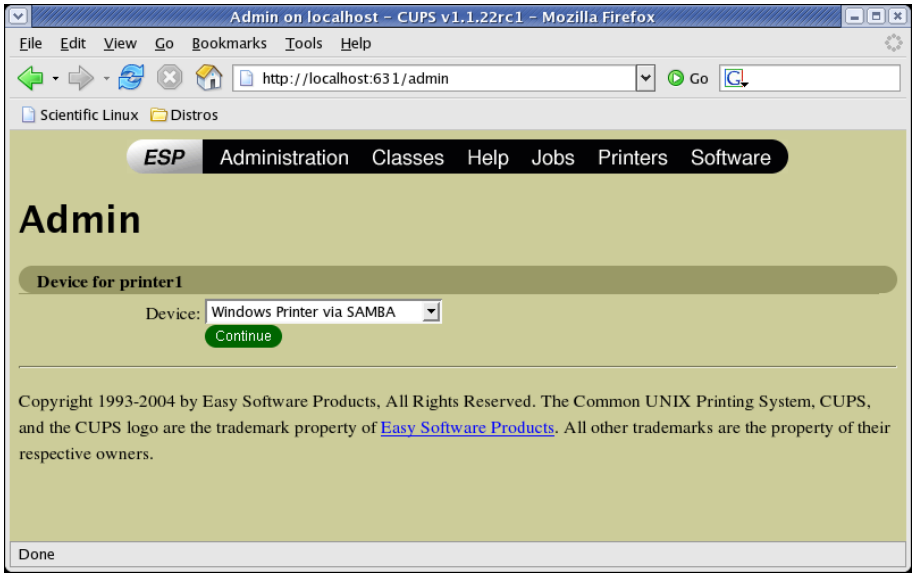

Рис. 11.5: Страница Модель/Драйвер веб-интерфейса CUPS

ВНИМАНИЕ

Использование имени пользователя и пароля при создании принтера может привести к вскрытию этих данных через файлы определений CUPS. Лучшей практикой является создание специального пользователя печати или открытие ресурсов печати для всех.

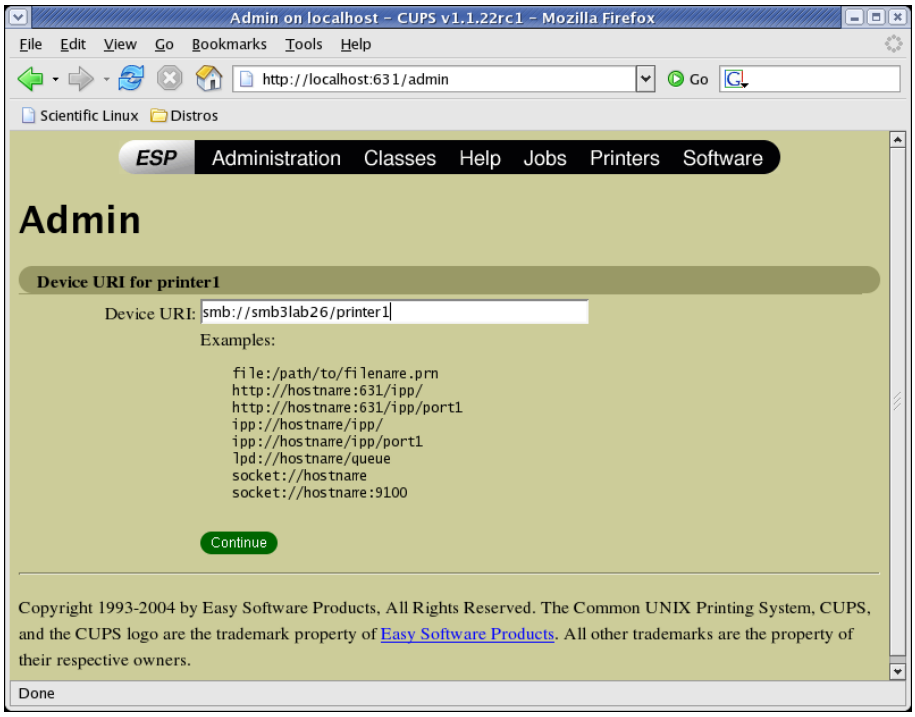

Рис. 11.6: Ввод URI устройства веб-интерфейса CUPS

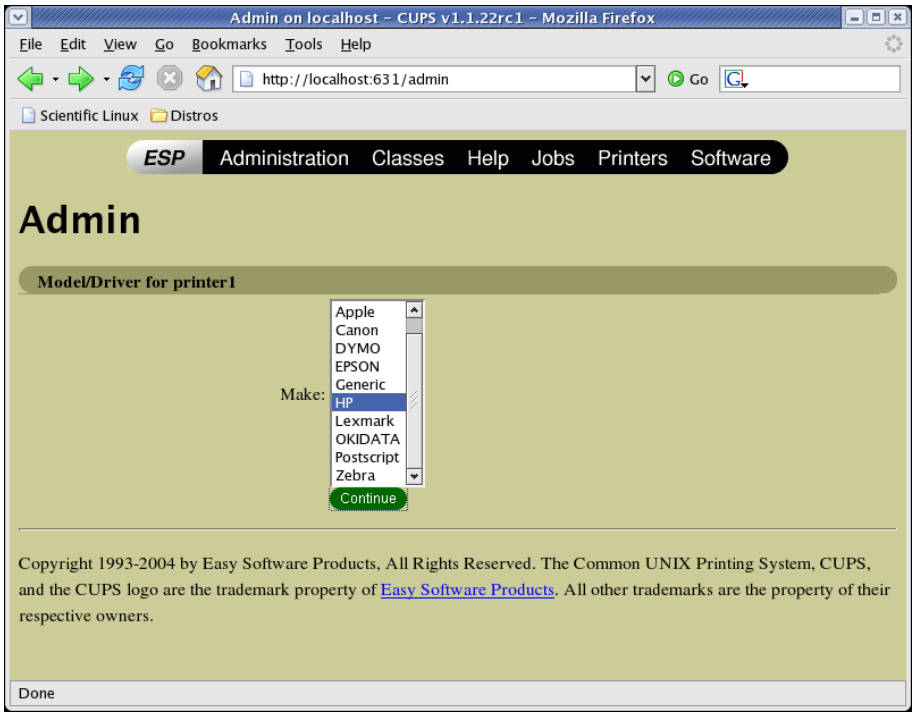

Рис. 11.7: Страница Модель/Драйвер веб-интерфейса CUPS

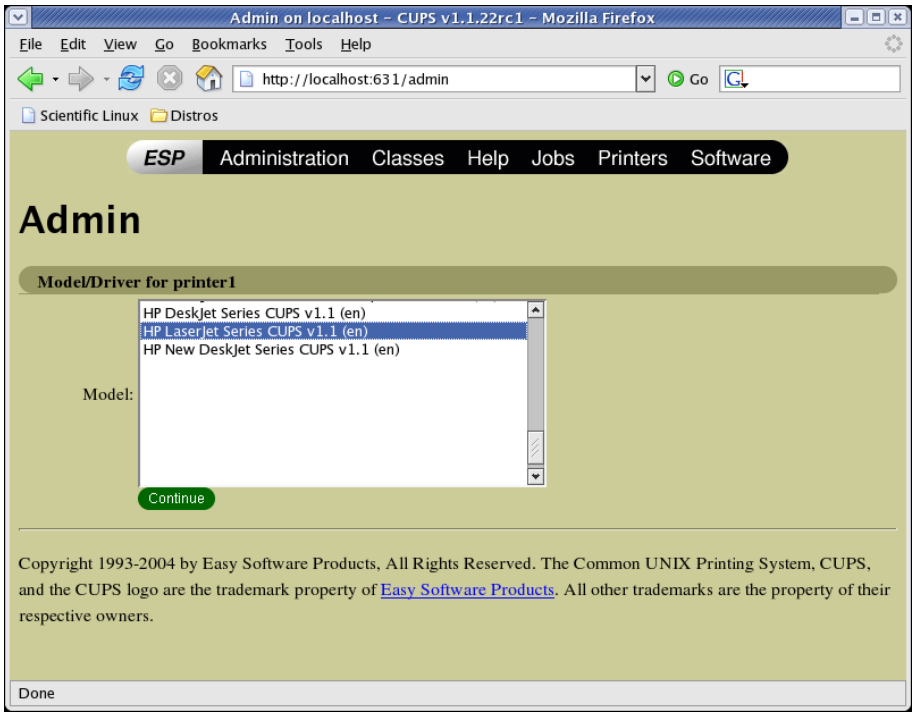

Рис. 11.8: Вторая страница Модель/Драйвер веб-интерфейса CUPS

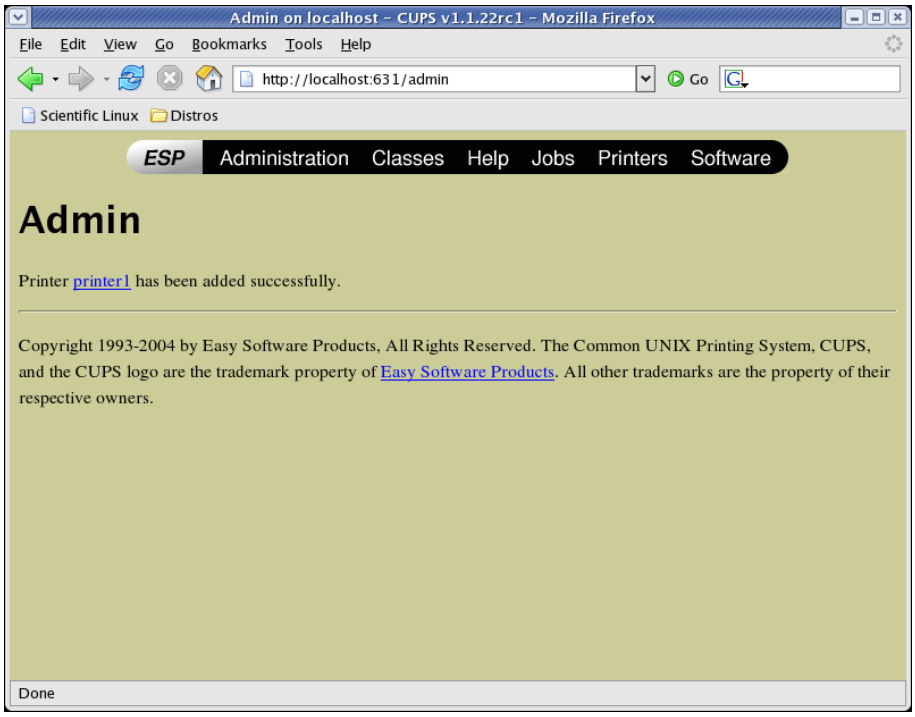

Рис. 11.9: Страница Принтер добавлен веб-интерфейса CUPS

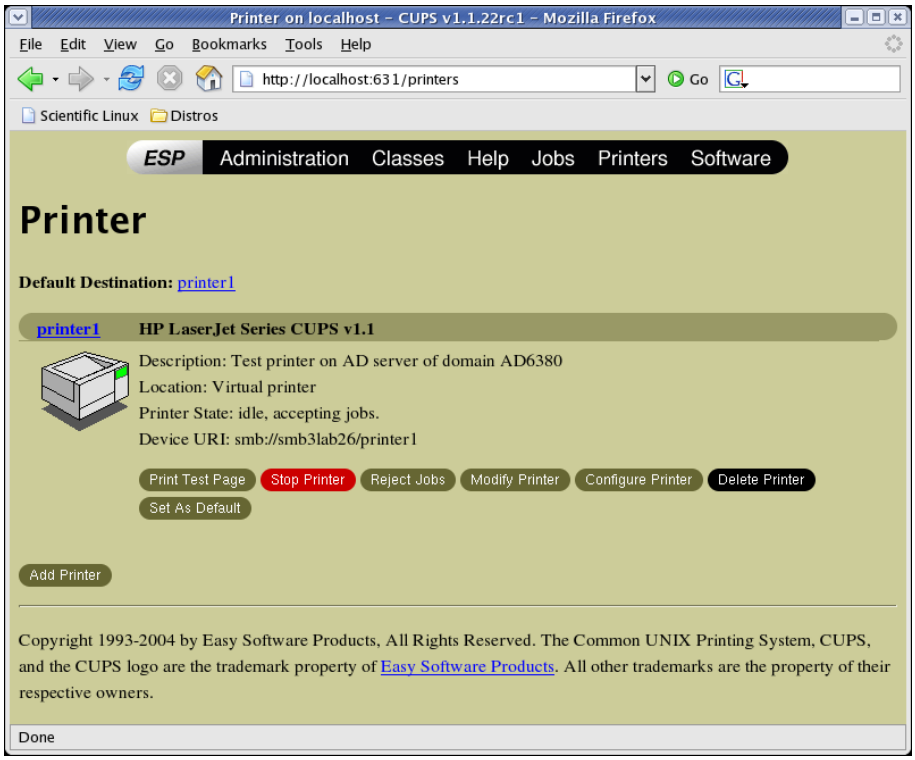

Рис. 11.10: Страница управления веб-интерфейса CUPS с добавленным принтером printer1

Часть V

Приложения

Приложение A

Словарь Linux для пользователей Windows

В этом приложении даются объяснения некоторых основных понятий...

A.1 Что все это значит?

Пользователь Microsoft Windows, впервые исследующий мир Linux, наверняка узнает множество новых терминов. Данный словарь бегло объясняет значение и роль многих терминов, акронимов и аббревиатур принятых для продуктов Linux. (Впрочем, некоторые из этих терминов характерны не только для Linux, но все же могут быть новыми для многих пользователей Windows.) Этот документ должен помочь в некоторой степени уменьшить путаницу вокруг Linux; однако большинство общепринятых терминов из сферы оборудования, программного обеспечения и коммуникаций в настоящий словарь не вошло, потому что их с легкостью можно найти в другом месте.

> ЗАМЕЧАНИЕ
>
> Некоторые из определений были упрощены с целью сделать их краткими. Предполагалось, что пособие просто даст новому пользователю общие разъяснения, а вовсе не станет всеобъемлющим учебником.

Если вы не можете найти нужное слово в этом словаре, попробуйте поискать его в других источниках. Существует множество словарей акронимов, аббревиатур и общих компьютерных терминов (не все они касаются исключительно Linux), среди которых можно выбирать. Поскольку некоторые термины могут присутствовать в одном словаре и отсутствовать в другом, объяснения в одном источнике могут быть более понятными и полными, в другом менее, здесь в алфавитном порядке приводится на выбор несколько источников:

- Free Online Dictionary of Computing

  http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?Free+On-line+Dictionary

- IBM Terminology web site

  http://www-306.ibm.com/software/globalization/terminology/

- Lucent Technologies Glossary

  http://www.lucent.com/search/glossary

- TechWeb TechEncyclopedia

  http://www.techweb.com/encyclopedia

- Ugeek Technical Glossary

http://www.geek.com/glossary/glossary.htm

- Whatis.com

  http://www.whatis.com

A.2 Наиболее распространенные термины Linux

Account Name (Регистрационное имя пользователя) То же, что и Login ID, User ID (ID пользователя) или User Name (Имя пользователя). Это имя, которое присваивается пользователю в системе Linux. Различные пользователи могут быть зарегистрированы в системе с уникальными именами, со своими уровнями доступа (правами). После установки Linux регистрационные имена пользователей назначаются Суперпользователем (Superuser), называемым также пользователем root.

ACPI — Advanced Configuration and Power Interface (Усовершенствованный интерфейс конфигурирования системы и управления энергопитанием) ACPI обеспечивает функциональность управления энергопитанием на x86 платформах.

ALSA — Advanced Linux Sound Architecture (Усовершенствованная звуковая архитектура Linux) Система доступа к звуковым картам и другим аудио-устройствам под Linux. ALSA включает в себя поддержку для наиболее популярных аудио-чипов и адаптеров. ALSA заменила OSS в большинстве дистрибутивов. См. также «OSS».

APM — Advanced Power Management™ (Усовершенствованные средства управления питанием) Промышленный стандарт, позволяющий системному процессору и различным компонентам переходить в энергосберегающий режим, включая режим приостановки, спящий режим и режим выключения. Программное обеспечение APM особенно важно для мобильных устройств, поскольку позволяет сохранять заряд батарей.

Archive (Архив) Один большой файл, содержащий несколько других, обычно сжатых для сохранения места на диске. Чаще всего архив создается, чтобы облегчить передачу данных на другой компьютер. Известные форматы архивов — arj, tar и zip.

ARJ Известная программа для сжатия и архивирования файлов, доступная в Linux, Windows и других операционных системах. Файлы, сжатые ею, обычно имеют расширение .arj или .ar.

Awk — Aho, Weinberger, and Kernighan Язык программирования, полезный своей возможностью использовать регулярные выражения, обычно применяется для восстановления и преобразования данных. GNU версия этой программы называется Gawk.

Background Process (Фоновый процесс) Программа, которая выполняется без пользовательского ввода. Несколько фоновых процессов может выполняться в многозадачной операционной системе, такой как Linux, в то время как пользователь взаимодействует с приоритетным процессом (например вводит данные). Некоторые фоновые процессы, например, демоны, никогда не требуют пользовательского ввода. Другие просто пребывают в фоновом режиме пока пользователь занят более приоритетной программой.

Bash — Bourne Again Shell Улучшенная версия Bourne Shell. См. также «Korn Shell».

BDF Fonts (Шрифты BDF) Ряд растровых шрифтов для системы X Window. См. также «PostScript Fonts». См. также «TrueType Fonts».

Bin Каталог, в котором находятся исполняемые программы, главным образом, бинарные файлы.

Binaries (Бинарники) Исходный код, который был скомпилирован в исполняемую программу. В мире Linux есть программное обеспечение, которое распространяется только в исходном коде, есть такое, которое включает в себя и исходники, и бинарники и есть программное обеспечение, распространяемое только в бинарном формате.

Boot Disk (Загрузочный диск) Диск (дискета или CD), на котором записана часть операционной системы (такой как Linux), достаточная для того чтобы загрузить компьютер и запустить некоторые необходимые программы из командной строки. Это может понадобиться, если система по какой-либо причине стала неспособной загружаться сама. Кроме всего прочего, загрузочный диск можно использовать для деления диска на разделы и его форматирования, восстановления Master Boot Record (главной загрузочной записи) или копирования определенных файлов.

Bootloader (Загрузчик) Приложение, которое управляет начальным запуском компьютера. Загрузчики устанавливают начальное окружение и затем передают процесс загрузки выбранной операционной системе. См. также «GRUB». См. также «LILO».

Bot Сокращение от Robot (Робот). Программа, разработанная для поиска информации в Интернете при минимальном участии человека.

Bourne Shell Известная оболочка командной строки с большим количеством преимуществ по сравнению с командной строкой в DOS. См. также «Bash». См. также «Korn Shell».

BSD UNIX (Berkeley Software Distribution UNIX) Дистрибутив UNIX из университета Калифорнии в Беркли. См. также «FreeBSD».

Bzip2 Более новая программа для сжатия файлов в Linux, обеспечивающая меньший размер файлов, чем Gzip. Расширение файлов обычно .bz2.

CGI — Common Gateway Interface (Общий шлюзовой интерфейс) Используется веб-серверами для обмена данными между сценариями или приложениями, а затем передачи данных веб-странице или броузеру. CGI сценарии обычно создаются на языке Perl могут генерировать динамическое содержимое веб-страницы (включая корзины заказов в электронной коммерции, форумы и анкеты).

CHS — Cylinder/Head/Sector (Цилиндр/Головка/Сектор) Информация о диске, которая требуется FDISK в процессе создания разделов.

Client (Клиент) Механизм, который запрашивает услуги (e-mail, например) у сервера.

Cluster (Кластер) Сеть из рабочих станций (PC или других) с Linux.

(См. также Beowulf.)

COLA Сокращение, которым можно ссылаться на Интернет-конференцию comp.os.linux.announce, где публикуются материалы, связанные с Linux.

Command Line Interface — CLI (Интерфейс командной строки) Текстовая сессия, полноэкранная или в виде окна, где пользователь запускает программы, набирая соответствующие команды с параметрами или без них. CLI показывает выходные данные операционной системы или программы и выдает приглашение командной строки для дальнейшего ввода пользователем данных.

Command Prompt (Приглашение командной строки) Термин DOS, Windows и OS/2®, которым называют часть командной строки, куда пользователь вводит команды. См. также «Shell Prompt».

Compile (Компилировать) Переводить исходный код программы в исполняемый.

Compiled Language (Транслируемый язык) Язык, который требует программу-компилятор, чтобы преобразовать исходный код в исполняемую программу в двоичном коде, понятную компьютеру. Программу достаточно скомпилировать один раз и впоследствии запускать на выполнение уже из полученного бинарника. Транслируемые языки и программы обычно быстрее, чем интерпретируемые или псевдокоды (p-code), но требуют дополнительного шага, компиляции, прежде чем приложение сможет быть запущено. Примеры транслируемых языков: C и C++, COBOL, FORTRAN.

Compiler (Компилятор) Программа, которая используется для преобразования исходного кода программы в исполняемый.

Console Application (Консольное приложение) Программа, которая выполняется из командной строки и не требует запуска графического пользовательского интерфейса (или возможно даже предлагает его запустить).

Cron Демон Linux, который выполняет определенные задания в назначенное время или интервал времени.

Daemon (Демон) Фоновый процесс операционной системы, обычно с правами доступа суперпользователя. Демон обычно прячется на заднем плане до тех пор, пока что-то (определенная дата, или время, или временной интервал) не активизирует его.

Desktop (Рабочий стол) Пользовательский интерфейс операционной системы, выполненный в виде офисного стола с различными предметами на нем. Вместо настоящих телефонных аппаратов, ламп и лотков для документов на рабочем столе операционной системы располагаются иконки различных программ и данных, окна, панели задач и тому подобное. В Linux доступно несколько различных окружений рабочего

стола, включая KDE и GNOME, которые пользователь может установить. См. также «GUI». См. также «Window Manager». См. также «X Window System».

Device Driver (Драйвер устройства) Программа, которая служит посредником между операционной системой и устройством (например, портами, дисководами, мониторами или принтерами). Она показывает операционной системе, какие возможности имеет устройство и переводит команды операционной системы в инструкции, понятные устройству.

Distribution (Дистрибутив) Предоставляемый пользователю комплект ядра Linux с различными пользовательскими интерфейсами, утилитами, драйверами и другим программным обеспечением. Обычно его можно бесплатно скачать или приобрести за умеренную плату комплект CD. Наиболее распространенные дистрибутивы — Red Hat, SUSE и Debian. Иногда дистрибутив называют «distro».

Dpkg — Debian Package Manager (Менеджер пакетов Debian) Встроенный в Debian Linux, но совместимый с другими дистрибутивами инструментарий для получения и установки пакетов через Интернет. Он создает файлы с расширением .deb. Похож на RPM.

Emacs — Editing with MACroS (Редактирование с помощью MACroS) Популярный текстовый редактор, обычно используется как консольное приложение. См. также «Vi».

Enlightenment («Просвещение») Один из нескольких пользовательских интерфейсов (менеджеров окон). Более подробно можно изучить на сайте http://www.enlightenment.org. См. также «GNOME». См. также «KDE». См. также «X Window System».

Environment variable (Переменная окружения) Переменная, которая используется в скриптах или консольных командах, ссылается на различные установки окружения. Часто встречающаяся переменная окружения $HOME, например, указывает на домашний каталог данного пользователя.

Executable bit (Бит исполнения) Часть файла, которая определяет, может ли файл быть выполнен непосредственно. Файлы без исполняемого бита считаются файлами данных. Обратите внимание, что обычно бывает возможно выполнить файл без такого бита, если использовать вспомогательное приложение. (Например, команда **perl ./noexec.pl** запустит приложение noexec.pl, даже если исполняемый бит в нем не установлен.)

File Extension (Расширение файла) Завершающие символы названия файла, расположенные после точки (.). Расширение файла обычно характеризует его тип. В отличие от Windows, в Linux исполняемые файлы обычно не имеют расширения.

File System (Файловая система) Набор программ, которые сообщают операционной системе как получить доступ и интерпретировать содержимое дисковода или лентопротяжного устройства или другого носителя информации. Распространенные файловые системы — FAT и NTFS в Windows и ext3 и ReiserFS в Linux.

Filter (Фильтр) Программа, которая считывает данные (из файла, вывода программы или из ввода командной строки), обрабатывает их в соответствии с заранее определенными условиями (например, сортирует в алфавитном порядке) и выводит обработанные данные. Некоторые фильтры включают в себя Awk, Grep, Sed и Sort.

Finger Команда Linux, которая предоставляет информацию о пользователях, зарегистрировавшихся в системе.

Foreground Process (Приоритетный процесс) В многозадачной операционной системе, такой как Linux, приоритетный процесс — это программа, с которой пользователь взаимодействует в данный момент (например, вводит данные). Разные программы могут становиться приоритетными в те моменты, когда пользователь переключается на них. В окружении с множеством окон, расположенных на разных уровнях, это соответствует самому верхнему окну.

FreeBSD — Free Berkeley Software Distribution Операционная система, которая, как и Linux, включает в себя множество GNU программ и позволяет использовать многие пакеты, аналогичные пакетам Linux. Однако некоторые функции ядра осуществляются здесь по-другому. См. также «BSD UNIX».

Fsck — File System Check (Проверка файловой системы) Команда для сканирования диска на наличие ошибок и по возможности исправления этих ошибок. Похожа на ScanDisk в Windows.

FTP — File Transfer Protocol (Протокол передачи файлов) Метод передачи файлов с и на другие компьютеры, зачастую являющиеся репозиториями программного обеспечения.

GCC — GNU C Compiler (GNU компилятор языка C) Высококачественный компилятор языка C, распространяемый под GPL.

GIMP — GNU Image Manipulation Program (GNU программа работы с изображениями) Популярный графический редактор в Linux.

GNOME — GNU Network Object Model Environment (Сетевая объектная среда GNU) Один из нескольких пользовательских интерфейсов (менеджеров окон) для Linux, построенный с помощью GTK+. Подробнее о GNOME читайте на сайте http://www.gnome.org.

При вербальном общении G не немой звук, как в слове Guh-Nome. См. также «Enlightenment». См. также «KDE». См. также «X Window System».

GNU Project — GNU is Not Unix (Проект GNU — GNU не есть Unix) Попытка Фонда Свободного ПО (Free Software Foundation — FSF) Массачусетского Института Технологии (Massachusetts Institute of Technology — MIT) разрабатывать и продвигать альтернативные запатентованным UNIX проекты. Программное обеспечение GNU попадает под лицензию GPL.

GNU/Linux То же самое, что Linux. Названо так, потому что многие компоненты, входящие в состав дистрибутива Linux, являются программами GNU.

GPL — GNU General Public License (Общая открытая лицензия GNU) Часто используемое лицензионное соглашение, регулирующее использование и распространение. Любая производная работы, выполненной под GPL, должна также выполняться под этой или похожей лицензией. Это относится к приложениям, которые взаимодействуют с библиотекой GPL. Копию соглашения GPL можно найти на странице http://www.gnu.org/copyleft/gpl.html.

Grep — Global Regular Expression® and Print Программа, которая ищет в файлах образец текста и выводит все строки, содержащие заданный образец.

GRUB — GRand Unified Bootloader Утилита управления загрузкой операционной системы. GRUB предоставляет графическое меню для выбора загружаемой операционной системы. GRUB заменил LILO во многих дистрибутивах. См. также «LILO».

GTK+ — GIMP ToolKit Мощная, быстрая, свободно распространяемая графическая библиотека для системы X Window в Linux, которую программисты используют для создания кнопок, меню и других графических объектов. При устном общении ее называют GTK. См. также «GNOME». См. также «Motif». См. также «Qt».

GUI — Graphical User Interface (Графический пользовательский интерфейс) Коллекция иконок, окон и других экранных графических изображений, позволяющих пользователю взаимодействовать с операционной системой. См. также «Desktop». См. также «Window Manager».

Gzip — GNU zip Используемая в Linux программа сжатия файла. Последние версии создают файлы с расширением .gz. (расширение .z или .Z показывает, что использовалась более старая версия Gzip.) Сжатие используется, чтобы уменьшить размер файла, тем самым сохраняется место на диске и уменьшается время передачи файла. (Когда Gzip используется вместе с Tar, результирующий файл имеет расширение .tgz, .tar.gz или .tar.Z.)

Hard Link (Жесткая ссылка) Жесткая ссылка — это связь между названием файла и его копией. При создании жесткой ссылки никакие данные не копируются, создается новый вход (новое имя) к исходному файлу. При удалении исходного файла жесткая ссылка, указывающая на него, остается. Жесткие ссылки могут указывать только на файлы внутри текущего разделе. См. также «Symbolic link».

Home Directory (Домашний каталог) Каталог, в который пользователь попадает сразу после входа в систему и в котором хранится большинство (если не все) его файлов. Обычно находится в подкаталоге домашнего каталога `/home`. Иногда на него можно ссылаться с помощью переменной окружения $HOME, которая указывает на домашний каталог данного пользователя. Также на него указывает специальный символ командного интерпретатора ~.

HTML — Hyper Text Markup Language (Язык разметки гипертекста) Стандартный язык разметки для проектирования веб-страниц. Теги разметки или команды форматирования позволяют разработчику веб-страниц выделять текст, определять расположение графики, создавать ссылки и т.д.

HTTP — Hyper Text Transport Protocol (Протокол передачи гипертекста) Набор указаний, созданный для запроса и отправки веб-страниц, основанных на применении языка разметки HTML.

Init Первый процесс, который запускается сразу после загрузки операционной системы. Он стартует систему в однопользовательском режиме или порождает оболочку для чтения стартовых файлов и открывает порты, обозначенные как порты входа.

Interpreted Language (Интерпретируемый язык) В отличие от транслируемой программы, которая один раз с помощью компилятора преобразуется из исходного кода в исполняемый файл, интерпретируемая программа преобразуется в бинарник «на лету» при каждом запуске программой-интерпретатором. Интерпретируемые языки (и, соответственно, написанные на них программы) обычно медленнее, чем транслируемые и p-code и в целом имеют ограниченные права доступа к низкоуровневым функциям операционной системы или прямого доступа к аппаратным ресурсам. С другой стороны, они часто включаются в операционную систему и обычно более просты для программирования, чем транслируемые языки. Примерами интерпретируемых языков могут служить BASIC, Perl, Python и REXX/Object REXX.

Java Объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems™ как язык, независимый от операционной системы. Java обычно используется на веб-серверах. Приложения и апплеты, написанные на Java, иногда предлагается загружать и запускать на пользовательских системах. На Java можно создавать и приложения, и более мелкие Java апплеты.

Хотя Java может быть компилирована в собственный код, обычно она транслируется в байткод, который затем интерпретируется. См. также «JIT».

Java Applets (Апплеты Java) Маленькие программы на Java, которые встроены в веб-страницу и выполняются броузером, в отличие от самостоятельных приложений. Апплеты не могут получать доступ к некоторым ресурсам (таким как файлы и принтеры) на локальном компьютере и, как правило, не могут общаться с другими компьютерами по сети.

JavaScript Межплатформный язык сценариев WWW, в какой-то мере имеющий отношение к Java. Он может использоваться как язык сценариев на стороне сервера, как язык, встроенный в HTML, обрабатываемый сервером, и как встроенный язык для броузеров.

JDK™ — Java Development Kit (Комплект разработчика для Java) Набор средств программирования на Java от компаний Sun, IBM или других, доступный в Linux и других операционных системах.

JFS — Journaled/Journaling File System (Журналируемая файловая система) Файловая система, в которую встроены возможности восстановления данных. Изменения в содержании, прежде чем вступить в силу, записываются в журнал (log-файл), поэтому если содержание повредится (например, из-за отключения питания в момент записи), его можно будет восстановить из журнала вместе с изменениями.

JIT Compiler (Компилятор JIT) Компилятор для интерпретируемых языков, позволяющий на лету автоматически компилировать программы в родной машинный язык для более быстрого представления программы. JIT расшифровывается как Just-In-Time — точно в срок.

Journaling (Журналирование) То же, что регистрация (logging). Запись информации в журнал (log-файл), как метод отслеживания изменений.

JVM™ — Java Virtual Machine (Виртуальная машина Java) Окружение Java, необходимое для запуска программ, написанных на Java, включающее в себя интерпретатор Java. Для каждой уникальной операционной системы (таких как Linux и Windows) требуется своя JVM, но любая JVM может запускать одну и ту же версию программы на Java.

KDE — K Desktop Environment Один из нескольких пользовательских интерфейсов (менеджеров окон) в Linux, собранный с помощью Qt. Более подробно о KDE можно узнать на сайте: http://www.kde.org. См. также «Enlightenment». См. также «GNOME». См. также «X Window System».

Kernel Ядро операционной системы, от которого зависят все остальные компоненты. Ядро управляет такими задачами, как низкоуровневое взаимодействие с аппаратными средствами и разделение ресурсов, включая распределение памяти, ввод/вывод, безопасность и доступ пользователей.

Korn Shell Улучшенная версия Bourne Shell, которая включает в себя мощную поддержку скриптов и редактирование командной строки. Поддерживает множество скриптов, написанных на Bourne Shell. См. также «Bash».

LGPL — Lesser GPL (Малая GPL) Вариант GPL, который обычно относится к программным библиотекам. Любое приложение может взаимодействовать с библиотекой или приложением, попадающими под LGPL, но любая производная библиотека должна попадать под LGPL или подобную ей лицензию. Копию лицензионного соглашения GPL можно найти на сайте: http://www.gnu.org/copyleft/lesser.html.

LILO — LInux LOader Утилита управления загрузкой раздела, способная загружать операционные системы. LILO создана не только для загрузки Linux. В большинстве дистрибутивов LILO заменено на GRUB. См. также «GRUB».

Link (Ссылка) Связь названия с файлом или каталогом. См. также «Symbolic link». См. также «Hard Link».

Linux UNIX-подобная операционная система с открытым кодом, изначально созданная Линусом Торвальдсом. Термин «Linux» на самом деле относится только к ядру операционной системы. Несколько сот человек внесло свой вклад в разработку ядра Linux. Оставшаяся часть дистрибутива Linux состоит из различных утилит, драйверов, приложений, пользовательского интерфейса и других программ, которые, в принципе, могут быть скомпилированы и запущены на других UNIX системах.

Log Служит для хранения приложений или системных сообщений или ошибок. Так же называется сам файл, в котором эта информация хранится.

Lynx Известный неграфический (текстовый) веб-броузер.

Macro (Макрос) Набор инструкций, хранящийся в исполняемой форме. Макросы бывают специфичны для конкретных приложений (такие как электронная таблица или текстовый процессор, которые делают определенные шаги внутри данной программы) либо «общего назначения» (например, макрос клавиатуры, который вставляет ID пользователя, при нажатии на клавиатуре комбинации Ctrl-U).

Man Команда Linux, вызывающая страницы справочного руководства.

MBR — Master Boot Record (Главная загрузочная запись) Первый физический сектор на загружаемом диске. Место, которое системный BIOS проверяет, чтобы определить какой из разделов является загрузочным (активным) и загрузить с него операционную систему при первоначальном включении компьютера.

Mesa Реализация OpenGL (Open Graphics Library — Открытая графическая библиотека) API (Application Programming Interface — Интерфейс программирования приложений). Mesa предоставляет стандартные правила и набор технических средств для написания 2D и 3D графического программного обеспечения с аппаратной поддержкой.

MIME — Multipurpose Internet Mail Exchange (Многоцелевые расширения почтовой службы в Интернете) Протокол передачи информации, который позволяет включать в текстовые сообщения электронной почты нетекстовые данные (например, графику, видео или аудио).

Motif Графическая библиотека в Linux, разработанная Фондом открытого программного обеспечения (Software Foundation — OSF), используемая программистами при создании кнопок, меню и других графических объектов для системы X Window. См. также «GTK+». См. также «Qt».

Mount (Монтировать) Сопоставлять диск файловой системе. Жесткими дисками, CD и дискетами можно пользоваться только после того, как они были подмонтированы. Извлекать диск из дисковода можно, как правило, после его отмонтирования. В большинстве современных дистрибутивов диски монтируются и отмонтируются автоматически. Также возможно подмонтировать образы дисков (такие как ISO-файлы), сетевые устройства и даже ссылки на другие части файловой системы.

Mount Point (Точка монтирования) Место в файловой системе, куда подмонтирован раздел жесткого диска, сменный носитель или какой-либо другой ресурс.

Multitasking (Многозадачность) Способность операционной системы выполнять более одной программы или задания одновременно. В невытесняющей многозадачной ОС подобной Windows 98 одно приложение должно освободить ресурсы по требованию, чтобы другое смогло использовать их. В вытесняющих многозадачных ОС, таких как Linux, Windows NT-подобных или OS/2, ресурсы освобождаются по запросу операционной системы по времени либо по приоритету так, что приложение не может захватить ресурсы, когда они нужны другой программе. См. также «Multithreading». См. также «Time-sharing».

Multithreading (Многопотоковость) Способность операционной системы одновременно запускать программы, которые были разделены на компоненты или потоки. Многопотоковость, при правильной организации, предлагает лучшее использование процессоров и других системных ресурсов. Текстовый процессор может удачно использовать многопотоковость, если будет проверять грамматику в приоритетном режиме и в это же время в фоновом сохранять на диск и печатать документ. См. также «Thread».

NFS — Network File System (Сетевая файловая система) Файловая система, которая дает доступ к файлам по сети или Интернету.

Newbie (Новичок) Некто, малознакомый с Интернетом, компьютерами вообще или Linux в частности (например, «Linux newbie»).

Object-Oriented (Объектно-ориентированный) Методология разработки программного обеспечения, которая предлагает программисту использовать уже написанные программные модули (компоненты), вместо того чтобы требовать от разработчика каждый раз писать свой программный код. Использование стандартных компонентов уменьшает время разработки (потому что написание и тестирование этих компонентов уже было проведено другими программистами) и улучшает стандартный вид и функции программ, использующих одни и те же модули.

OO См. «Object-Oriented».

Open Source (Открытый исходник) Несколько неясный термин, относящийся к программному обеспечению, которое выпускается вместе со своим исходным кодом. Факт присутствия исходного кода еще не означает, что пользователи могут изменять и распространять его. Этот термин иногда подменяют выражением «Свободное ПО», хотя это не всегда одно и то же. См. также «Public Domain». См. также «Shareware».

OSS — Open Sound System (Открытая звуковая система) Драйвер для доступа к звуковым картам и другим аудио-устройствам под Linux. Он «вырос» из Linux Sound Driver и поддерживает наиболее известные аудио-чипы и адаптеры. В большинстве дистрибутивов OSS заменен на ALSA. См. также «ALSA».

OSS — Open Source Software (ПО с открытым исходным кодом) См. «Open Source».

Owner (Владелец) Пользователь с привилегированными правами доступа к файлу; обычно тот, кто этот файл и создал.

P-code (Pseudo-code) Language (П-код, псевдокод) Род интерпретируемого языка. Псевдокод по способу исполнения программ на нем написанных представляет из себя что-то вроде гибрида транслируемых и интерпретируемых языков. Как и в интерпретируемых языках, программы на псевдокоде переводятся в бинарный вид автоматически при запуске. Однако, в отличие от того, как это происходит при использовании транслируемых языков, исполняемый бинарный файл хранится в псевдокоде. Вдобавок, в отличие от интерпретируемого языка, программу не надо переводить в бинарник при каждом запуске. После того как она первый раз конвертируется в исполняемый псевдокод, при каждом дополнительном выполнении используется именно он, если только в начальной программе не были внесены изменения. Псевдокод (и программы, написанные на нем) обычно медленнее, чем транслируемые языки и программы, но быстрее, чем интерпретируемые. Кроме того, как правило, имеет права доступа к некоторым низкоуровневым функциям операционной системы, хотя не имеет прямого доступа к аппаратным средствам. Такие языки обычно встроены в операционные системы и некоторые из них проще транслируемых. Примерами псевдокода могут служить Java, Python и REXX/Object REXX.

PAM — Pluggable Authentication Modules (Подключаемый модуль аутентификации) Сменный модуль идентификации пользователя для системы безопасности, который позволяет писать программы, не используя информацию о том, какая схема аутентификации будет использоваться. Благодаря этому, со временем, при замене одного модуля другим не придется переписывать программы.

Panel (Панель) В Linux так называют то, что соответствует Панели задач (Taskbar) в Windows.

Partition (Раздел) Непрерывный участок диска, который рассматривается операционной системой, как физический диск. Таким образом, один диск может иметь несколько точек монтирования, приписанных ему.

PCF fonts (Шрифты PCF) Множество растровых шрифтов для использования в системе X Window.

PDF (Portable Document Format) files (Файлы PDF) Бинарные файлы, которые создаются с помощью программы Adobe Acrobat или других программ, способных формировать этот формат на выходе. Используются для создания платформно-независимых документов, которые можно просматривать с помощью Acrobat Reader или других программ, включая веб-броузеры с подключенным Acrobat Reader.

Perl — Practical Extraction and Report Language (Язык Perl) Широко известный язык программирования. Часто используется на веб-серверах Linux для создания CGI-скриптов.

Permission (Право доступа) Полномочие читать и писать в файлы и каталоги и выполнять программы. Различные уровни доступа назначаются Суперпользователем отдельным файлам, каталогам либо пользователю (User ID). Права доступа указываются одной из двух форм сокращенной записи. При этом права доступа разбиты на три группы: первая — доступ для владельца файла, вторая — доступ для группы и третья — доступ для всех пользователей. В первой форме сокращения записи используется аббревиатура

прав доступа, тогда rwx означает read (читать) write (писать) и execute (выполнять). Более общая форма сокращения записи использует для обозначения прав доступа числа. Число соответствует сумме прав доступа, считается, что read (читать) равно четырем, write (писать) — двум и execute (выполнять) — одному. К примеру, права доступа к файлу, владелец которого может читать и писать в него, группа только читать, а все остальные вообще не имеют доступа к нему, будут записаны как 640.

PGP — Pretty Good Privacy Высокосекретная, использующая схему с открытыми ключами программа шифрования данных в Linux и других операционных системах.

Pipe (Канал) См. «|».

Port (Перенос) Процесс, когда программу, написанную для одной операционной системы, подправляют, чтобы с теми же функциональными возможностями выполнить в другой ОС. Для адаптации программы к конкретным особенностям новой ОС зачастую может потребоваться незначительная модификация, в отличие от оптимизации приложения под конкретную ОС.

Portable (Переносимый) Термин относится к программному обеспечению, разработанному для использования в более чем одной операционной системе с минимумом исправлений и перекомпиляцией.

POSIX — Portable Operating System Interface for uniX (Интерфейс переносимой операционной системы для uniX) Набор стандартов программного интерфейса, который регулирует как написать исходный код приложения, чтобы оно было переносимо между операционными системами. POSIX базируется на UNIX и служит основой для спецификации X/Open, созданной компанией The Open Group.

PostScript Язык описания страниц, разработанный фирмой Adobe Systems, который сообщает принтеру, как отображать текст и графику на печатаемой странице.

PostScript Fonts (Шрифты PostScript) Огромное множество шрифтов, которые можно использовать в OS/2, Microsoft Windows и системе X Window.

Файлы шрифтов могут иметь расширения .afm, .pfa и .pfb. Иногда эти шрифты называют шрифты AdobeType 1 или шрифты ATM (Adobe Type Manager — менеджер шрифтов Adobe). Обычно для шрифтов PostScript требуется PostScript принтер.

Process (Процесс) Исполняемая программа. См. также «Multitasking». См. также «Multithreading».

Public Domain (Общедоступное ПО) Программное обеспечение, которое может быть использовано и изменено любым человеком в любых целях, его даже можно включать и распространять в составе коммерческого ПО. Общедоступное ПО не защищено никакими авторскими правами. См. также «Open Source». См. также «Shareware».

Public Key Encryption (Шифрование с открытым ключом) Метод шифрования данных, в котором используется пара ключей: открытый и секретный. Данные, зашифрованные с открытым ключом, можно расшифровать только с помощью секретного и наоборот. Обычно открытый ключ сообщается и может быть использован для расшифровки данных, переданных владельцем секретного ключа, а секретный ключ используется в качестве подписи.

Python (Питон) Объектно-ориентированный язык программирования, псевдокод.

Qt Мощная, быстрая графическая библиотека с открытым кодом для системы X Window в Linux, используется программистами для создания кнопок, меню и других графических объектов. В разговоре произносится так же как и слово «cute» («кьют», привлекательный). См. также «GTK+». См. также «KDE».

Queue (Очередь) Список задач, ожидающих исполнения, например, «очередь печати» («the print queue»).

RAID — Redundant Array of Independent/Inexpensive Disks/Devices (Избыточный массив независимых/недорогих дисков/устройств) Метод обеспечения избыточности данных, улучшенной работы и быстрого восстановления их при сбоях в работе диска путем разделения или дублирования данных на несколько дисков. Часто используемые типы RAID — это RAID 0 (Data Striping, расщепление данных), RAID 1 (Disk Mirroring, зеркалирование диска) и RAID 5 (Striping with Distributed Parity, расщепление с распределенным контролем по четности). Конфигурации RAID обычно требуют, чтобы диски были идентичны (одинаковой емкости и даже бренда и модели). Массивы RAID воспринимаются операционной системой, как одно устройство.

RC File (Файл RC) Файл сценария, содержащий стартовые инструкции для программы (приложения или даже операционной системы). Файл, который должен автоматически выполняться, когда начинает за-

гружаться операционная система, содержит список инструкций (команд или других сценариев), которые надо выполнить.

RCS — Revision Control System (Система контроля изменений) Набор программ, который контролирует разделяемый доступ к файлам в группе и отслеживает изменения в тексте файла. Обычно используется для обслуживания программирования исходных кодов модулей.

Rdev Утилита для получения информации о системе Linux. Используется, чтобы запросить и установить загрузочное устройство, видео-режим, устройство подкачки и диск RAM.

Root User (Пользователь root) ID пользователя, которому разрешено выполнять задачи всех системных уровней. См. также «Superuser».

Root Window (Корневое окно) Основная сессия, в которой запущен рабочий стол Linux.

RPM — RPM Package Manager (Менеджер пакетов RPM) Инструментарий для получения и установки пакетов (в том числе через Интернет, включен в некоторые дистрибутивы Linux. Он создает файлы с расширением .rpm. Похож на Dpkg.

Script (Сценарий или скрипт) Набор команд, хранящихся в файле. Используется для автоматического, повторяющегося исполнения процессов. См. также «RC File».

SCP — Secure Copy (Безопасное копирование) Метод для безопасного копирования файлов между локальной и удаленной машиной. SCP использует SSH как backend. См. также «SSH».

Session (Сессия) Полный период взаимодействия пользователя с операционной системой, начиная от его входа в систему и заканчивая выходом.

Shareware (Условно-бесплатное ПО) Вид коммерческого программного обеспечения, в котором предлагается «испытать прежде чем купить». Если заказчик продолжает пользоваться продуктом после короткого испытательного периода, его просят заплатить определенную, обычно номинальную, плату. См. также «Open Source». См. также «Public Domain».

Shell (Командная оболочка) Текстовое окно, в котором содержится интерфейс командной строки операционной системы.

Shell Prompt (Приглашение оболочки командной строки) Область командной строки, куда пользователь вводит данные. Приглашение командной строки в DOS выглядит как символ «Больше» (>), а в Linux обычно как «Процент» (%), «Доллар» ($) или какой-нибудь другой в зависимости от того, какая командная оболочка используется. См. также «Command Prompt».

Shell Script (Shell-скрипт) Сценарий, разработанный для автоматического выполнения при запуске командной оболочки.

Slash (Косая черта) Название символа /. Используется в указании пути к файлу, в отличие от обратной косой линии (\), принятой в операционных системах DOS, Windows и OS/2.

Source Code (Исходный код) Операторы языка программирования, записанные программистом в текстовом файле. В некоторых языках программирования команды могут выполняться на лету программным интерпретатором. В других языках необходимо компилировать команды в исполняемые программы (бинарники), а уже затем выполнять. В мире Linux есть программное обеспечение, которое распространяется только в исходном коде, есть такое, которое включает в себя и исходники, и бинарники и есть программное обеспечение, распространяемое только в бинарном формате.

Spool — Simultaneous Peripheral Operation On-Line (Одновременные операции с периферийным оборудованием в реальном времени) Отправлять данные программе, которая копит информацию, для более позднего использования, например, очередь печати (print spooler).

SQL — Structured Query Language (Язык структурированных запросов) Язык, который используется для того, чтобы управлять записями и полями (рядами и колонками) в реляционных базах данных. Иногда произносится как «сиквел» («sequel»).

SSH — Secure Shell Безопасный протокол, позволяющий входить на удаленную машину. Многие протоколы могут быть «туннелированы» в SSH, так что обмен между двумя машинами будет происходить через SSH (в зашифрованном виде), а не в открытую по сети.

String (Строка) Последовательность символов, как в «строке поиска».

Superuser (Суперпользователь) Обычно синоним для пользователя root.

Swap (Подкачивать) Временно перемещать данные (программы или файлы данных) из оперативной памяти в память диска (swap out, откачивать) или обратно (swap in, подкачивать), чтобы могло обрабатываться больше программ и данных, чем может позволить размер физической памяти. Это явление также называют виртуальной памятью.

Swap Space (Область подкачки, область свопинга) Место на диске, где временно хранятся подкачиваемые данные. Linux использует для области подкачки специальный раздел диска вместо файла подкачки.

Symbolic link (Символьная ссылка) Ссылка на файл или каталог. В английском иногда сокращают до «symlink». Если файл удаляется, символьная ссылка на него перестает работать. См. также «Hard Link».

Sync (Синхронизировать) Загружать весь ожидающий ввод/вывод данных на диск.

Syslog Linux System Logger (системный регистратор Linux), где хранятся все сообщения или ошибки системы.

Tag (Тег) Команда языка разметки, такого как HTML, предписывающая отображать информацию в определенном виде, например, по центру страницы, полужирным шрифтом или с использованием определенного шрифта.

Tar — Tape ARchive (Архив на магнитных лентах) Входящая в состав Linux программа компоновки файлов, которая собирает несколько файлов в один для облегчения его дальнейшего архивирования. Изначально она была разработана для создания резервных копий на магнитных лентах, но сегодня может использоваться на любом другом носителе информации. Сама по себе она создает файлы с расширением .tar, а вместе с программой сжатия данных Gzip — .tgz, .tar.gz или .tar.Z.

Tarball Файл, создаваемый утилитой Tar, в котором содержится один и более других, архивированных и, возможно, сжатых, файлов.

TeX Известная программа форматирования текстов, использующая макросы. Является основой программ LaTeX и teTeX.

Text Editor (Текстовый редактор) Программа редактирования текстовых файлов. Похожа на текстовый процессор, но не имеющая большинства функций форматирования (таких как границы, курсив и шрифты). Обычно используется для написания и редактирования скриптов, программ и простых текстовых файлов.

Text Formatter (Программа форматирования текстов) Программа, которая готовит текстовый документ к печати, позволяет пользователю задать множество функций разметки, таких как границы, колонтитулы, отступы, деление на страницы и выравнивание.

TFTP — Trivial File Transfer Protocol (Тривиальный протокол передачи файлов) Упрощенная версия FTP — без аутентификации и многих других основных возможностей FTP.

Thread (Поток) Небольшая часть программы, которая действует как независимое подмножество большой программы. Многопотоковая программа может работать гораздо быстрее монолитной или однопотоковой, так как разные задачи в ней выполняются одновременно. Кроме того, потоки внутри одного приложения могут разделять ресурсы и передавать данные друг другу.

Time-sharing (Разделение времени) Метод организации доступа к процессору многих пользователей путем выделения для каждого из них части процессорных ресурсов на какой-то сегмент времени. См. также «Multitasking».

Torvalds, Linus (Торвальдс, Линус) Создал ядро Linux в 1991, является держателем авторских прав на Linux и в настоящее время остается координатором проекта развития Linux.

Touch Команда, которая меняет дату и время создания (или последнего изменения) файла, не затрагивая его содержимое. Если файла с заданным названием не существует, то будет создан пустой файл.

TrueType Fonts (Шрифты TrueType) Большое множество шрифтов, которые, в отличие от шрифтов PostScript не зависят от принтера. См. также «BDF Fonts». См. также «PostScript Fonts».

Tux (Тукс) Имя вымышленного пингвина — талисмана Linux.

Umount Команда для отмонтирования раздела жесткого диска, сменного носителя информации или других ресурсов. См. также «Mount».

UNIX UNIX начинался как собственная операционная система корпорации Bell Laboratories, разработанная ею в 1960-х. Со временем появилось множество несовместимых друг с другом коммерческих версий от

компаний Apple (Mac OS X), Digital (Digital UNIX), Hewlett-Packard (HPUX), IBM (AIX®), NeXT (NeXTSTEP) и других.

UUCP Набор программ и протоколов, которые стали основой для международной сети UNIX-компьютеров. Расшифровывается как UNIX to UNIX Copy Program (Протокол связи UNIX-машин).

Vi Известный текстовый редактор, который обычно используется как консольное приложение. Существует более новая версия Vim. См. также «Emacs».

Virtual Desktop (Виртуальный рабочий стол) Метод, благодаря которому рабочее пространство пользователя расширяется за пределы экрана компьютера. Рабочий стол можно прокручивать влево и вправо, вверх и вниз, как будто за стеклом экрана расположен большего размера рабочий стол, и передвигая его, можно увидеть не поместившиеся иконки, окна и другие объекты. В качестве альтернативы, как это сделано в KDE, могут быть доступны несколько кнопок, по каждой из которых показывается область рабочего стола со своими объектами, по размеру равная реальному экрану монитора.

Virtual Machine (Виртуальная машина) Виртуальные машины (VM) — функции центральной микросхемы процессора, которые отделяют область памяти от остальной системы. Поскольку операционные системы и приложения работают в защищенном режиме, то, если программа зависнет на одной виртуальной машине, это не повлияет на деятельность программ и операционных систем вне этой виртуальной машины.

Virtual Memory (Виртуальная память) Процесс использования части дискового пространства в качестве временного хранилища памяти. Синоним Swap.

Widget (Виджет) Программный объект графического пользовательского интерфейса (такой как кнопка, полоса прокрутки, флажок) в системе X Window. См. также «X Window System».

Window Manager (Менеджер окон) Графический пользовательский интерфейс (GUI), который работает в системе X Window и дает пользователю доступ к окнам, иконкам, панелям управления и другим объектам рабочего стола. См. также «Desktop».

Working Directory (Рабочий каталог) Другое название для текущего каталога или каталога, в котором пользователь работает в настоящий момент времени.

Workspace (Рабочая область) Другое название для корневого окна (Root Window) или рабочего стола (Desktop).

Wrapper Программа, которая используется для запуска другой программы.

X Window System (Система X Window) Графическое окружение с многооконным интерфейсом для UNIX. Основная программа, необходимая многим пользовательским интерфейсам. См. также «Desktop». См. также «Window Manager».

X11 11-ая версия системы X Window.

XDM — X Display Manager (X менеджер экранов) Пользовательский дружественный внешний интерфейс входа в систему X Window.

XML — eXtensible Markup Language (Расширяемый язык разметки) Мощный новый язык разметки для оформления данных; похож на HTML, но позволяет программистам самим определять теги разметки и команды форматирования.

Zip Известная форма сжатия и архивирования файлов, доступная в разных операционных системах, включая Windows и Linux. Включает в себя программы PKZip/PKUnzip и Zip/Unzip. Файлы, созданные такими программами имеют расширение .zip.

> Символ перенаправления; обычно используется, чтобы направить результат действия программы в текстовый файл. Например, следующая команда перенаправляет содержание текущего каталога в файл `output.txt`, стирая при этом то, что было записано в этом файле раньше:

```
ls -a > output.txt
```

См. также «>>». См. также «|».

>> Символ добавления; обычно он используется, чтобы направить результат действия программы в текстовый файл, дописывая данные в конец, а не заменяя содержимое файла. Например, следующая команда записывает содержание текущего каталога в конец файла `output.txt.`:

```
ls -a >> output.txt
```

Повторный вызов этой команды продолжит запись в конец файла. См. также «>». См. также «|».

| Символ «канал» (на типовой 101-клавишной клавиатуре символ Shift-Backslash, расположенный над клавишей Enter); обычно используется, чтобы направить результат действия одной программы на вход другой. Например, чтобы найти в результате предыдущей команды слово «mcopy», используется следующая команда:

```
history | grep mcopy
```

См. также «>>». См. также «>».

/ См. «Slash».

~ См. «Home Directory».

Приложение B

Персональные настройки пользователя

Это приложение посвящено сравнению способов создания и хранения персональных настроек пользователя в Windows и в Linux. Поскольку в фокусе нашей книги находится миграция именно клиентского ПО, то на вопросы, связанные в персональными настройками пользователя, следует обратить особое внимание как на этапе планирования, так и на этапе реализации проекта миграции.

B.1 Персональные данные пользователя в Windows

Набор персональных данных пользователя в Windows называют профилем (profile). В Windows 2000 и Windows XP данные пользовательского профиля хранятся в каталоге:

```
c:\Documents and Settings\username
```

В Windows NT все собрано в каталог:

```
c:\winnt\profiles\username
```

В Windows эти каталоги используются для хранения различных видов индивидуальных данных, например, пользовательских документов, настроек и данных, относящихся к приложениям, временных файлов, файлов «предпочтений» и т.д. Имя каталога, как правило, соответствует имени пользователя, но может содержать и другие детали, например, имя компьютера или домена. Такие каталоги возникают в том случае, если создается новый пользователь, но с таким именем, которое уже существует на компьютере, для интеграции в домен. Такая практика приводит к тому, что на одном компьютере может быть несколько различных пользователей с одинаковым именем, но разными «добавками» к имени, например, может быть пользователь username.localdomainname и username.localcomputername. Этих пользователей надо различать.

Выбор профиля выполняется во время авторизации. Данные, которые определяют настройки рабочего стола для данного пользователя и некоторые другие установки, хранятся в файле `NTUSER.dat`. После ввода пароля эти данные загружаются, благодаря этому пользователь видит привычное для него рабочее окружение.

Такой тип профилей хранится локально и доступен только на конкретной машине. Если пользователь с тем же именем открывает сессию на другой машине, то там для него создаются отдельные файлы профиля.

В том случае, если авторизация пользователя не привязана строго к одной машине, и есть необходимость хранить персональные данные не локально, а на сетевых ресурсах, то в Windows используется другой тип профиля, называемый роуминг-профиль.

Третьим типом профилей в Windows является профиль mandatory. Этот профиль применяется в тем случаях, когда хотят запретить пользователю вносить изменения в свои настройки. В этом случае редактировать данные профайла может только администратор системы, и все изменения, возникшие во время сессии, уничтожаются при выходе из нее.

В.2 Персональные данные пользователя в Linux

Теперь рассмотрим, как хранятся персональные данные в Linux. Поскольку Linux не представляет собой, в отличие от Windows, монолита, а имеет модульную структуру, то данные настроек не могут храниться в единой базе-регистре, как в Windows. В Linux принято хранить данные настроек в легко читаемых и редактируемых текстовых файлах, имеющих расширения `.conf` или `.profile`.

В файле `/etc/profile` содержатся общесистемные настройки и информация о программах, которые запускаются во время авторизации. Настройки конкретного пользователя, которые расширяют или замещают системные, как правило, находятся в домашнем каталоге пользователя, который определен в переменной `$HOME`. Стандартным домашним каталогом для всех дистрибутивов Linux является

`/home/username`

Имя файла с настройками может быть различным в различных дистрибутивах. Например, в Red Hat Enterprise Linux локальный профиль хранится в файле `.bash_profile` (Bourne Again SHell), а в SuSE Linux используется файл `.profile`.

Поскольку файлы настроек являются текстовыми файлами в формате ASCII, то они могут быть прочитаны без применения специальных программ. Другим достоинством такого метода хранения настроек является их доступность для автоматического администрирования. Для установки ограничений на модификацию настроек можно использовать как стандартные средства операционной системы, то есть права доступа, так и дополнительные.

В Linux многие приложения для хранения своих конфигурационных файлов создают скрытые каталоги, имена которых начинаются с точки.

Способы хранения и поддержания персональных данных пользователя, касающихся настройки рабочего окружения, зависят от выбора менеджера рабочего стола. Ниже мы рассмотрим способы хранения персональных данных в двух наиболее популярных менеджерах: KDE и GNOME. Вопросы, которые мы будем обсуждать, уже частично рассматривались в Разд. 7.3.

В.3 Персональные настройки для рабочего стола под управлением KDE

Настройки KDE базируются на персональных данных пользователя, которые хранятся в подкаталоге домашнего каталога пользователя

`/home/username/Desktop`

Этот каталог содержит иконки и ссылки на каталоги, которые представлены на рабочем столе.

Для хранения всех конфигурационных файлов в KDE используется простой текстовый формат ASCII, их легко редактировать любым текстовым редактором и обрабатывать с помощью скриптов. Ниже приведен список основных конфигурационных файлов:

- `kdeglobals`
- `kdesktoprc`
- `kickerrc`

В дистрибутиве Red Hat эти файлы хранятся в каталоге `/usr/share/config`. В других дистрибутивах место расположения их может быть другим. Эти файлы содержат персональные настройки, которые приписываются каждому новому пользователю. Таким образом, все изменения, сделанные в этих файлах, будут унаследованы вновь создаваемыми пользователями. Некоторые моменты, связанные с управлением этими файлами, уже обсуждались в Разд. 9.1.

В.4 Персональные настройки для рабочего стола под управлением GNOME

Так же как и в случае с KDE, персональные настройки рабочего стола под управлением GNOME хранятся в подкаталоге домашнего каталога пользователя. Но структура файлов настройки GNOME отличается от структуры файлов настройки KDE. Информация о персональных настройках хранится в каталоге:

`/home/username/.gnome-desktop`

Работа с иконками в GNOME организована так же как и в KDE. То есть, информация о них хранится в простых текстовых файлах в формате ASCII, в них же определяется поведение приложения, связанного с иконкой. Новые файлы иконок могут быть сохранены пользователем в каталоге `/home/<user>/Desktop`.

Однако, для настройки общей конфигурации пользователя в GNOME применяются совсем другие форматы. Для настройки GNOME используется база данных в формате XML. Файлов настройки достаточно много и расположены они в каталоге

`/etc/gconf/gconf.xml.defaults/desktop/gnome/`

Каждому виду настроек соответствует свой подкаталог, ниже приведен пример структуры каталога настройки:

```
/etc/gconf/gconf.xml.defaults/desktop/gnome/accessibility
/etc/gconf/gconf.xml.defaults/desktop/gnome/applications
/etc/gconf/gconf.xml.defaults/desktop/gnome/background
/etc/gconf/gconf.xml.defaults/desktop/gnome/file_views
/etc/gconf/gconf.xml.defaults/desktop/gnome/font_rendering
/etc/gconf/gconf.xml.defaults/desktop/gnome/interface
/etc/gconf/gconf.xml.defaults/desktop/gnome/peripherals
/etc/gconf/gconf.xml.defaults/desktop/gnome/sound
/etc/gconf/gconf.xml.defaults/desktop/gnome/thumbnailers
/etc/gconf/gconf.xml.defaults/desktop/gnome/url-handlers
```

Внутри каждого их этих каталогов должен находиться файл с именем `%gconf.xml`.

Для внесения изменений в XML-файлы настроек GNOME применяется инструмент gconftool-2, который вызывается из командной строки.

Для актуализации настроек используется демон gconfd-2, этот демон оповещает обо всех внесенных в настройки изменениях те приложения и утилиты GNOME Desktop, которые используют эти настройки, а также оповещает об изменениях другие инструментальные программы менеджера GNOME. Для каждого пользователя GNOME запускается свой экземпляр демона.

Некоторые моменты, связанные с техникой изменения настроек конфигурации GNOME, и вопросы ограничения доступа пользователей и групп к этим настройкам уже обсуждались в Разд. 9.2.

Приложение C

Управление рабочим столом и скрипты

В этом приложении рассматриваются вопросы применения скриптов для управления рабочим столом под Linux и языки программирования, ориентированные на написание скриптов. Примеры, которые мы рассматриваем, могут быть полезны как для конечных пользователей, так и для администраторов системы.

C.1 Языки, ориентированные на создание скриптов

Существует несколько языков программирования, ориентированных на создание скриптов и доступных под Linux, среди них Perl, Python, Ruby, ECMAScript и Tcl. Они разрабатывались для UNIX/Linux-подобных систем, но постепенно распространились и были перенесены под другие операционные системы, такие как Windows и Mac OS X. Основной язык скриптов системы Windows, Visual Basic, не имеет прямого аналога в Linux, однако и для Linux существуют проекты, аналогичные проекту Windows Visual Basic for Applications (VBA). Некоторые из них будут упоминаться ниже. Например, существует коммерческий продукт REALbasic и свободная разработка StarBasic, см.:

http://en.wikipedia.org/wiki/StarOffice_Basic

Информацию о свободных разработках можно найти в сети по адресам:

http://www.thefreecountry.com/compilers/basic.shtml

http://documentation.openoffice.org/HOW_TO/various_topics/VbaStarBasicXref.pdf

Эти диалекты Basic полезны в некоторых случаях, например, при импорте файлов из Excel, но не играют большой роли в автоматизации управления настольной машиной.

C.1.1 Язык сценариев и командный интерпретатор

Традиционный способ управления системой под Linux (как и под UNIX) состоит в использовании сценариев (или скриптов), обрабатываемых командными интерпретаторами. Примерами интерпретаторов могут служить bash, ksh, tcsh. Скрипты Linux в некоторой степени подобны `.bat`-файлам Windows, но командные интерпретаторы обладают значительно большими возможностями. Скрипты используются при начальной загрузке системы (загрузочные скрипты можно найти в `/etc/init.d`), а также в качестве специальной оболочки для запуска более сложных приложений, таких как Mozilla или OpenOffice.org. Общую информацию о командах оболочки и их интерпретаторах можно найти по адресам:

- *Linux Shell Scripting Tutorial:*

  http://www.freeos.com/guides/lsst

- *Advanced Bash-Scripting Guide:*

  http://www.tldp.org/LDP/abs/html/

C.1.2 Perl

Perl (Practical Extraction and Report Language) — это один из самых мощных и гибких языков программирования, ориентированных на создание скриптов. Первоначально он проектировался и разрабатывался как язык обработки текстов, но постепенно стал использоваться и для других задач. Существует огромное число модулей, написанных на Perl, их можно найти на сайте CPAN (Comprehensive Perl Archive Network) по адресу http://www.cpan.org. Энтузиасты этого языка любят его за то, что на нем легко программировать, вокруг программирования на Perl возникла философия «TMTOWTDI» («There's More Than One Way To Do It»). Многие, однако, считают, что Perl не подходит для решения значительных по объему задач, поскольку тексты на Perl трудно читать и сопровождать.

C.1.3 Python

Python — современный объектно-ориентированный язык. Он достаточно мощный и обладает ясным, хорошо читаемым синтаксисом. Благодаря большому объему встроенных библиотек, обеспечивает абстрагирование для многих платформо-ориентированных функций, таких как доступ к файлам и обработка потоков. Есть много библиотек для графических приложений, таких как PyGTK, PyQt, wxPython и даже средства для работы с Mac OS X и Microsoft Foundation Classes (MFC). Модули, использующие эти библиотеки, могут быть написаны на Python, C или C++. На Python могут быть также написаны и большие самостоятельные приложения, но, главным образом, он используется как вспомогательный язык, для расширения возможностей приложений, например, на нем пишут плагины.

Red Hat интенсивно использует Python во время процедуры установки системы с дистрибутива и в некоторых других случаях. На Python написан такой инструмент распознавания аппаратного обеспечения как kudzu, который используется также и в Debian-дистрибутиве Knoppix. Для определения характеристик аппаратного обеспечения можно просто выполнить команду **kudzu-probe**, но Python позволяет администратору системы написать более элегантные скрипты для распознавания определенных видов оборудования, пример такого скрипта приведен ниже.

Пример C.1.1: Диагностика аппаратного обеспечения с использованием kudzu и Python

```
#!/usr/bin/env python

import kudzu

print kudzu.probe(kudzu.CLASS_SOCKET, kudzu.BUS_PCI, kudzu.PROBE_ALL)
print kudzu.probe(kudzu.CLASS_NETWORK,kudzu.BUS_PCI, kudzu.PROBE_ALL)

devices = kudzu.probe(kudzu.CLASS_UNSPEC, kudzu.BUS_UNSPEC, kudzu.PROBE_ALL)
for dev in devices: print dev
```

C.1.4 Встроенные языки

Большинство языков, ориентированных на разработку скриптов, позволяют писать простые графические приложения или апплеты рабочего стола, существуют скриптовые оболочки для библиотек GTK+, Qt и wxWidgets и даже для KDE/GNOME API. Некоторые приложения имеют собственные встроенные языки главным образом для того, чтобы дать возможность пользователю запрограммировать какие-то специфические действия. В некоторых случаях в качестве встроенных используются общие языки. Например, Perl и Python используются в Gnumeric и в GIMP, JavaScript и XUL используются в Mozilla и связанных с Mozilla приложениях (таких как Firefox и Thunderbird), Star Basic используется в OpenOffice.org.

C.1.4.1 Qt Script

Фирма Trolltech выпустила специальный инструмент QSA (Qt Script for Applications), который обеспечивает поддержку сценариев в приложениях, написанных на C++. QSA включает интерпретатор для ECMAScript

(стандартизованный вариант JavaScript). Программы, написанные с использованием Qt Script, имеют полный доступ к интерфейсу прикладного программирования (API) Qt. Подробнее о QSA можно прочесть на странице Trolltech:

http://www.trolltech.com/products/qsa/

QSA играет ту же роль, что и VBA для Windows-приложений, его очень легко использовать любому, кто знаком с Qt. QSA распространяется под коммерческой лицензией для платформ Linux, Windows и Mac. Кроме того, QSA можно получить под лицензией GNU GPL для разработок с открытым кодом.

C.1.4.2 KJSEmbed

Подобный подход реализован в инструменте KJSEmbed, который предоставляет удобную оболочку для KDE-интерпретатора ECMAScript (kjs). Эта библиотека обеспечивает связывание между Qt/KDE-приложениями и KJE с использованием протокола DCOP. KJSEmbed распространяется под лицензией LGPL, полную документацию и сам инструмент можно найти в сети по адресам:

http://xmelegance.org/kjsembed

http://www.sourcextreme.com/presentations/KJSEmbed/

Пример C.1.2: Простой HTML броузер с использованием KJSEmbed

```
#!/usr/bin/env kjscmd

var url = 'http://www.kde.org/';
if (application.args.length)
   url = application.args[0];

var mw = new KMainWindow();
var box = new QHBox(mw);
mw.setCentralWidget(box);

var view = Factory.createROPart("text/html", box, "view");
view.connect(view.child(0),
             'openURLRequest(const KURL&,const KParts::URLArgs&)',
             'openURL(const KURL&)')
view.openURL(url);

mw.resize(650,500);
mw.show()
application.exec();
```

C.1.5 Kommander

Создатели Quanta+, популярного HTML-редактора для KDE, разработали новый инструмент, который называется Kommander. Этот инструмент позволяет разрабатывать диалоги и виджеты для KDE, используя программу kmdr-editor (изначально основывавшийся на Qt Designer), и затем выполнять их с помощью программы kmdr-executor. Kommander использует описание GUI-элементов в формате XML и встроенные DCOP-скрипты, скрипты могут быть написаны на любом языке, который имеет поддержку этого протокола (Bash, Python, Perl, Java или JavaScript). Kommander позволяет писать диалоги для KDE с непосредственным использованием механизмов DCOP. Эта возможность неоднократно использована внутри Quanta+. Ниже на Рис. C.1 Показан пример интегрированной DCOP-разработки с помощью программы kmdr-editor. Более подробную информацию о Kommander можно найти в сети по адресам:

http://kommander.kdewebdev.org

http://quanta.kdewebdev.org

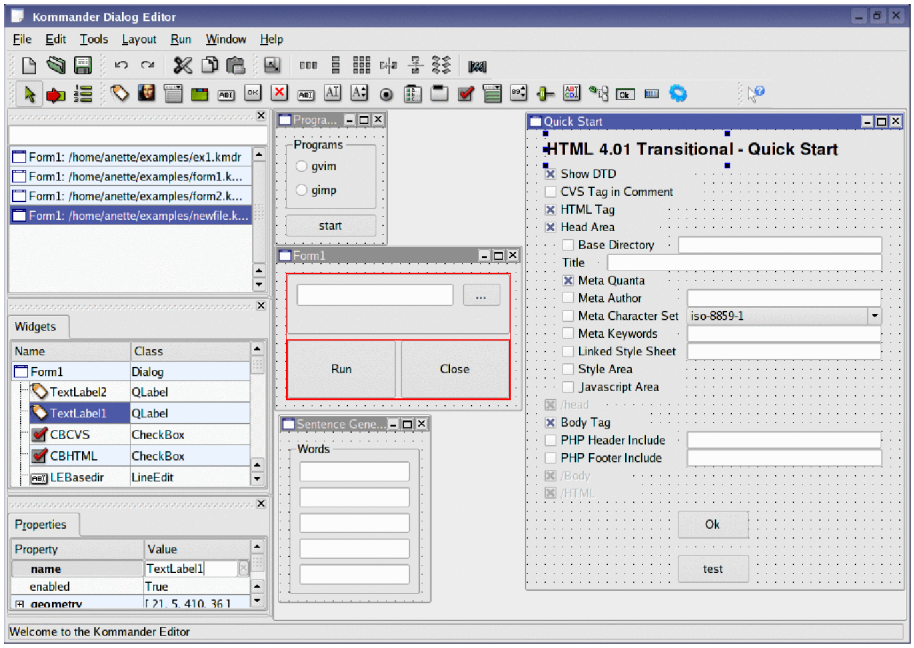

Рис. C.1: Интегрированная DCOP-разработка с помощью Kommander

C.2 Обмен данными между процессами

Многие приложения для рабочего стола Linux поддерживают различные средства обмена данными между процессами, что позволяет контролировать поведение приложений программно. Также полезно бывает использовать обмен данными между процессами в том случае, если есть желание использовать один язык для управления всеми приложениями, а не пользоваться встроенным языком сценариев, для каждой программы своим.

Разработчики приложений для GNOME поначалу использовали ORBit2 (оптимизированный вариант CORBA), но затем перешли на D-BUS.

Для KDE стандартным протоколом обмена данными между процессами является DCOP. Он базируется на протоколе ICE (Inter Client Exchange) и использует сокеты вместо удаленных вызовов. Этот протокол первоначально являлся вариантом CORBA, но затем был полностью переделан под нужды взаимодействия процессов рабочего окружения, поскольку скорость обработки в рамках CORBA была слишком низкой. DCOP можно применять также и для работы с KDE-приложениями на рабочем столе под управлением GNOME. Ниже мы рассматриваем этот продукт более подробно.

C.2.1 DCOP

Когда вы открываете свою сессию под KDE, то по команде **startkde** запускается программа kdeinit, эта программа стартует другие приложения, такие как dcopserver и KDE-демон kded, которые отвечают за управление системной конфигурацией и кэширование (объединение этих функций сокращенно называют syscoca), за поддержку обмена данными между процессами и многое другое. Если вы хотите узнать подробнее, как они работают, и какие переменные окружения и конфигурационные файлы влияют на их работу, прочтите руководство по KDE для системного администратора, которое можно найти в сети по адресу:

http://www.kde.org/areas/sysadmin/

Вы можете анализировать рабочее окружение и выполнять вызовы DCOP при помощи инструмента командной строки **dcop**, или из его графической оболочки kdcop, или с помощью DCOP-вызовов из таких языков как Perl, Python, Ruby, Java. Синтаксис командной строки **dcop** следующий:

```
dcop [ application [object [function [arg1] [arg2] [arg3] ... ] ] ]
```

C.2.2 Автоматизация рабочего стола с применением DCOP

Для того чтобы начать работать с DCOP, просто выполните команду **dcop**, проверьте ее вывод и перенаправьте его другому процессу. Ниже, в Прим. C.2.1 показано, как это работает. Сначала мы получаем список всех приложений, зарегистрированных на DCOP-сервере (некоторые из них включают ID, например, процесс консоли имеет ID равный 366), затем открываем новую консольную сессию и показываем имя первой сессии. После этого выводится список всех учетных записей электронной почты, проверяется наличие почты, производится сжатие почтовых папок и открывается окно для создания нового сообщения, при этом используются некоторые предустановленные значения.

Пример C.2.1: Работа с DCOP

```
% dcop
konsole-366
kmail

...
% dcop konsole-366 konsole newSession
% dcop konsole-366 session-1 sessionName
% dcop kmail KMailIface accounts

% dcop kmail default checkMail
% dcop kmail default compactAllFolders
% dcop kmail default openComposer info@kde.org "" "" "Subject" "DCOP \
   Rocks" 0
```

Следующий пример посвящен работе с броузером. Мы открываем сессию Konqueror с предпочтительным URL, затем открывается еще одно окно броузера с двумя закладками и предустановленными URL. Затем Konqueror переводится в полноэкранный режим, что может быть полезно для презентаций и работе в режиме Kiosk. В последней строке мы переходим на виртуальный рабочий стол номер 2, используя публичный интерфейс менеджера окон KDE kwin.

Пример C.2.2: Еще один пример работы с DCOP

```
% dcop
konqueror-21209
kwin
...
% dcop konqueror-21209 konqueror-mainwindow#1 openURL \
   http://www.ibm.com/linux
% dcop konqueror-21209 default openBrowserWindow http://www.kde.org
% dcop konqueror-21692 konqueror-mainwindow#2 newTab \
   http://www.linux.com

% dcop `dcop konqueror-21209 konqueror-mainwindow#1 action fullscreen` \
   activate

% dcop kwin default setCurrentDesktop 2
```

DCOP обладает богатыми возможностями, его можно использовать даже в комбинации с XML-RPC (XML Remote Procedure Calls) посредством моста XML-RPC — DCOP. Используя стандартные меры обеспечения безопасности UNIX при работе с ICE-сокетами, можно быть уверенным в том, что ни один пользователь не сможет вмешаться в работу DCOP-сессии другого пользователя.

Приложение D

Перенос приложений

На этапе планирования миграции, проводя инвентаризацию программного обеспечения, вы могли обнаружить, что среди используемого ПО есть приложения, которые не могут работать под Linux. К таким приложениям, например, относятся все программы, написанные с использованием Microsoft Visual Basic.

Это приложение содержит информацию и ссылки на ресурсы в сети, которые помогут узнать больше о процессе переноса приложений между платформами. Перенос приложений — это очень большая тема, ей одной можно было бы посвятить целую книгу. Мы ограничимся только введением в тему и перечислим некоторые проекты, с ней связанные.

Обширное руководство по планированию и выполнению переноса приложений можно найти в сети на странице фирмы Novell:

http://developer.novell.com/wiki/index.php/Porting_and_Migration

D.1 GTK+

GTK+ представляет собой кросс-платформенный инструментарий для разработки графических приложений, GTK+ используется такими программами как The GIMP и GAIM и менеджером рабочего стола GNOME. Он написан на языке C, обращения к нему можно писать не только на C, но и на C++, Python, Perl и других языках программирования. GTK+ распространяется под лицензией LGPL, его можно использовать как в разработках с открытым кодом, так и в коммерческих продуктах. Подробную информацию о GTK+ можно найти в сети по адресу:

http://www.gtk.org

D.2 Qt

Qt также является набором кросс-платформенных графических инструментов, этот инструментарий разработан фирмой Trolltech. Qt используется менеджером рабочего стола KDE и связанными с ним приложениями. Qt написан на языке C++, обращения к нему можно писать также и на C, Python, Ruby и других языках программирования. Qt имеет две лицензии — GPL и коммерческую лицензию, таким образом его можно бесплатно использовать в разработках с открытым кодом, но для коммерческого использования необходимо приобрести лицензию у фирмы Trolltech. Подробную информацию о Qt можно найти в сети по адресу:

http://www.trolltech.com

D.3 REALBasic

Продукт компании REAL Software Inc., REALBasic разработан как кросс-платформенное средство разработки, обеспечивающее эффективный путь переноса приложений, написанных с использованием Visual Basic, на плат-

форму Linux. Подробное описание процесса переноса приложений с использованием REALBasic можно найти на страницах технической документации фирмы REAL Software Inc. в сети по адресу:

http://www.realbasic.com/vb

D.4 wxWidgets

wxWidgets (ранее известная как wxWindows) — это кросс-платформенная библиотека инструментов с открытым исходным кодом для построения графического интерфейса пользователя (GUI). wxWidgets позволяет собирать приложения для Win32, Mac OS X, GDK+, X11 и других систем. Библиотека написана на C++, но может подключаться ко множеству других распространённых языков, таких, как Python (wxPython), Smalltalk (wxSqueak), Perl и Java. wxWidgets распространяется под лицензией wxWidgets License, и может быть бесплатно использована как в разработках с открытым кодом, так и в коммерческих приложениях. Подробную информацию о wxWidgets можно найти в сети по адресам:

http://www.wxwidgets.org

http://en.wikipedia.org/wiki/WxWidgets

На страницах фирмы IBM можно познакомиться с двумя детальными руководствами по использованию этого инструмента:

- *Looking through wxWindows: An introduction to the portable C++ and Python GUI toolkit*:

  http://www-106.ibm.com/developerworks/library/l-wxwin.html

- *Porting MFC Applications to Linux: A step-by-step guide to using wxWindows*:

  http://www-106.ibm.com/developerworks/linux/library/l-mfc

D.5 Mono и .NET Framework

Mono — проект по созданию полноценной реализации платформы .NET фирмы Microsoft на базе свободного программного обеспечения. Mono базируется на спецификациях ECMA для двух языков программирования: языка платформы .NET CIL (Common Intermediate Language) и языка C#. Большинство стандартных библиотек платформы .NET реализовано в рамках Mono. Однако в версии Mono 1.1 возможности графической библиотеки WinForms, которая используется большинством приложений платформы .NET, поддерживаются только частично. Работы по полной реализации библиотеки WinForms ведутся довольно энергично, ожидается, что полная поддержка будет включена в версии 1.2. Mono также включает поддержку GTK#, графического инструментария, который работает как под Linux, так и под Windows.

Mono включает компилятор языка C#, а также могут выполняться модули, написанные на языках Visual Basic.NET, IronPython, Nemerle и Boo. Языки, которые работают в среде CLR (Common Language Runtime) имеют доступ ко всем ресурсам платформы и могут гладко взаимодействовать с другими языками даже в рамках одного приложения.

Некоторые приложения платформы .NET могут содержать специфические вызовы API, но большинство существующих приложений для .NET легко могут быть перенесены под Mono. Подробную информацию о Mono можно найти в сети по адресу:

http://www.go-mono.com

Руководство для тех, кто хочет научиться переносить приложения, написанные для платформы .Net, под Mono, можно найти по адресу:

http://developer.novell.com/wiki/index.php/Porting_and_Migration

Приложение E

Обзор текущего состояния ПО с открытым кодом

E.1 Базовое программное обеспечение

Базовое ПО, включает в себя набор программных средств, которые используются постоянно. Их конкретный состав может различаться в зависимости от сферы применения. Основными компонентами, входящими в состав большей части таких наборов, являются ядро операционной системы, базовый набор системных утилит.

Базовым компонентом таких комплексов является ядро операционной системы, обеспечивающее нижний уровень взаимодействия оборудования, программных компонент и пользователей. Наибольшее распространение в мире Open Source технологий получили операционные системы, построенные по идеологии ОС UNIX, хорошо зарекомендовавшей себя в ходе более чем 30-летней истории эксплуатации в различных прикладных областях. Наиболее распространенные из них:

1. Linux
2. BSD-подобные ОС (FreeBSD, NetBSD, OpenBSD)
3. OpenSolaris
4. Darwin

Безусловным лидером по динамике развития, распространенности, удобству использования и поддержке сторонних производителей является ОС Linux.

Стандартом де факто в качестве базового набора системных утилит, отвечающих промышленному стандарту POSIX, является набор программ проекта GNU, используемых практически во всех свободных UNIX-подобных системах.

E.2 Сообщества разработчиков

Большинство проектов по созданию ПО с открытым кодом ведется сообществами разработчиков, так называемыми community. Некоторые из них спонсируются государственными или коммерческими организациями или специальными фондами. Каждый проект координируется модератором, который определяет направления дальнейшего развития проекта, регулирует состав и функциональную нагрузку членов сообщества, определяет модераторов отдельных направлений и подпроектов. Различные члены сообщества могут заниматься различными функциями: разработкой, тестированием, переводами, составлением документации, дизайном, поддержанием работоспособности инфраструктуры разработки. Наиболее известные Open Source проекты:

- Linux kernel
- BSD-подобные ОС (FreeBSD, NetBSD, OpenBSD)
- GNU

- X-Window
- GNOME
- KDE

Наиболее известные ресурсы, на которых ведутся OpenSource проекты и аккумулируются сведения о них:

- Проект GNU (http://www.gnu.org)
- SourceForge (http://www.sourceforge.net)
- Savannah (http://savannah.nongnu.org)
- Gna! (http://www.gna.org)

Наиболее известные информационные ресурсы, на которых публикуются новости, связанные с проектами OpenSource:

- FreshMeat (http://freshmeat.net)
- Slashdot.org (http://slashdot.org)
- Linux Center (http://www.linuxcenter.ru)
- LOR (http://linux.org.ru)
- Nixp (http://www.nixp.ru)

E.3 Дистрибутивы

Дистрибутивы Linux условно можно разделить на серверные, клиентские и универсальные. Большинство из наиболее распространенных дистрибутивов являются дистрибутивами универсальными, которые служат базой для создания специализированных дистрибутивов.

Первые дистрибутивы Linux создавались сообществами энтузиастов и распространялись на безвозмездной основе. Поддержка осуществлялась через то же сообщество. Многие из таких проектов живут и развиваются и по сей день. Главным достоинством этого рода дистрибутивов является нулевая цена приобретения, дающая возможность приобщаться к ним всем желающим, в том числе и молодежи, являющейся наиболее активной частью движения Open Source. Главными недостатками, присущими данным проектам является сложность процесса установки и настройки таких систем для неподготовленных пользователей и непредсказуемость динамики развития таких проектов, целиком зависящая от энтузиазма конкретных членов сообщества. Эти недостатки служат серьезным препятствием для внедрения таких систем в организациях, не специализирующихся в области работы с программным обеспечением.

Наиболее популярные дистрибутивы, поддерживаемые сообществами разработчиков:

- Slackware: старейший дистрибутив из сохранившихся на сегодняшний день, послуживший базой для создания многих современных дистрибутивов. Главным недостатком является архаичность организации, в первую очередь подсистемы обслуживания программных пакетов, что не позволяет поддерживать большие программно-аппаратные комплексы.
- Debian: наиболее распространенный дистрибутив из этого разряда. На нем базируется большое число вторичных программных проектов, таких как KNOPPIX, Ubuntu и др. Главным недостатком является большой промежуток между официальными релизами, занимающий зачастую несколько лет. В условиях быстро меняющегося в мире Open Source технологического ландшафта этот недостаток может иметь критическое значение.
- Gentoo: один из самых молодых и динамично развивающихся проектов. Ориентирован на оптимизацию программных пакетов путем их непосредственной сборки на рабочей машине. Очень удобен для целей оперативного слежения за текущим состоянием программных продуктов, входящих в дистрибутив, но для использования в организационной структуре, критичной к стабильности работы, не рекомендуется.
- Fedora Core: программный проект по созданию свободного экспериментального дистрибутива для разработчиков, поддерживаемый сообществом разработчиков и спонсируемый компанией Red Hat. Многие из решений, отработанных на этой тестовой платформе, инкорпорируются впоследствии в продукты Red

Hat Enterprise Linux. По пути создания таких же экспериментальных дистрибутивов пошли и другие поставщики программных продуктов, такие как Novell (Open Suse), Sun Microsystems (Open Solaris).

Коммерческие дистрибутивы представляют из себя продукты так называемого Enterprise уровня, предназначенные для работы в сферах, где устойчивость работы является основным требованием. Для таких дистрибутивов гарантируется поддержка производителя в течении 3-5 лет после выпуска очередной версии. Главными недостатками являются высокая стоимость приобретения данных продуктов и зависимость от коммерческой и технологической политики конкретного коммерческого производителя.

Наиболее известные коммерческие производители дистрибутивов:

- Red Hat: наиболее старый из них, безусловный технологический лидер. Компания разработала систему пакетов RPM, ставшую на сегодняшний день стандартом для производителей коммерческих дистрибутивов. Основной продукт на сегодняшний день — это линейка дистрибутивов Red Hat Enterprise Linux, все исходные пакеты из состава дистрибутива и обновлений доступны для свободного использования. Это породило ряд проектов по пересборке этих пакетов и созданию свободно-распространяемых дистрибутивов Enterprise уровня: Scientific Linux, White Box, CentOS, TAO и др.
- Novell (Suse)
- Mandriva

В последнее время намечается отчетливая тенденция по созданию региональных дистрибутивов, лучше отвечающих местным культурным и технологическим реалиям, например Asianux (Китай, Япония, Южная Корея).

В России хорошо известны дистрибутивы отечественных разработчиков, таких как Alt Linux, ASP Linux, Линукс Инк и др.

E.4 Репозитории

Крайне важными в практике использования ПО с открытым кодом можно считать создание неких банков, аккумулирующих пакеты программного обеспечения, оптимизированные и оттестированные под конкретные дистрибутивы.

Примеры репозиториев, связанных с конкретными дистрибутивами:

- Debian
- Gentoo
- Fedora
- Sisyphus (Alt Linux) и др.

Репозитории, содержащие наборы пакетов под различные дистрибутивы:

1. Dag Wieers — http://dag.wieers.com/home-made/apt
2. ATrpms (Axel Thimm) — http://atrpms.net
3. Dries Verachtert — http://dries.ulyssis.org/rpm
4. Linux Ink Cyrillic Edition — ftp://ftp.linux-ink.ru

E.5 Портальные технологии

Интенсивно развивающиеся в последнее время WEB-технологии минимизируют затраты на развертывание и внедрение информационных систем. Продукты, созданные с использованием таких технологий, являются кроссплатформенными и не требуют установки дополнительного ПО на рабочие места пользователей. При этом обеспечивается возможность создания «толстых» клиентов для построения сложных пользовательских приложений и создания интерфейсов для взаимодействия с внешними программными комплексами.

Исторически самыми первыми системами, предоставляющими сайты с динамическим формированием WEB-страниц, были специализированные системы, написанные на различных языках программирования (C, Perl, Python, LISP и др.) и работающие через CGI-интерфейс готовых WEB-серверов или же сами предоставляющие

HTTP-сервис. Некоторые из таких систем предоставляют вполне достаточную для своей области применения функциональность и продолжают развиваться и по сей день.

Организация порталов с парадигмой целостного пользовательского окружения, механизмами авторизации и разграничения прав пользователей, поддержкой типизации объектов и возможностью задания логики изменения их состояния (workflow) такими средствами на сегодняшний день представляется весьма затруднительной, если вообще возможной.

Наиболее распространенными средами для создания таких систем порталов с возможностью управления содержимым сайта, так называемыми Content Management System (CMS), на настоящий момент являются среды, созданные на основе технологий PHP, Java и Zope.

E.5.1 PHP

Одна из наиболее старых технологий создания динамических сайтов. Обзор и обсуждение портальных решений на базе технологий PHP можно найти на сайте:

http://www.opensourcecms.com/

Наиболее популярные Open Source проекты, использующие данные технологии:

- Drupal (http://drupal.org)
- Mambo (http://mamboserver.com)
- PHPNuke (http://www.phpnuke.org)

Данная технология имеет достаточно широкую распространенность, полностью открыта, достаточно проста в изучении и не требовательна к аппаратным ресурсам.

Отрицательными её чертами можно признать достаточно низкоуровневую организацию самих средств разработки, отсутствие стандартных механизмов для поддержки workflow, разграничения прав доступа и, вытекающие отсюда, проблемы с обеспечением безопасности.

Данные технологии слабо приспособлены для создания больших масштабируемых проектов, но вполне удовлетворяют потребностям при создании небольших WEB-сайтов (особенно если местные специалисты уже владеют данными технологиями), не очень критичными с точки зрения обеспечения безопасности.

E.5.2 Java

Обзор и обсуждение портальных решений на базе технологий Java можно найти на сайте:

http://java-source.net/open-source/content-managment-systems

Наиболее популярные Open Source проекты, использующие данные технологии:

- OpenCMS (http://www.opencms.org)
- Apache Lenya (http://lenya.apache.org)
- Alfresco (http://www.alfresco.org)

Данная технология является промышленным стандартом в сфере разработки ПО, поддерживается мировыми лидерами ИТ-индустрии: IBM, Sun Microsystems, Microsoft, BEA и др. и позволяет взаимодействовать с программными средами, созданными согласно спецификации J2EE. Данный подход позволяет строить большие масштабируемые решения.

Недостатками данных технологий являются закрытость процесса разработки данной платформы в рамках одной коммерческой компании, неполная открытость исходных кодов, высокий порог вхождения в данные технологии, являющийся следствием их сложности, сложность интеграции с другими платформами разработки, отсутствие критической массы готовых модулей верхнего уровня и решений на этой платформе, позволяющих выполнять быстрое прототипирование приложений пользовательского уровня.

Данная платформа обычно позиционируется как средство для создания инфраструктурных составляющих в ИТ-проектах для решения крупных задач.

E.5.3 Zope

Обзор и обсуждение портальных решений на базе технологий Zope можно найти на сайте Zope:

http://www.zope.org/

Наиболее популярные Open Source проекты, использующие данные технологии:

- CMF (http://www.plone.org)
- Plone (http://www.plone.org)
- CPS (http://www.cps-project.org)
- Silva (http://www.infrae.com/download/Silva)

Наибольшее развитие и распространение получила технология Plone. Данная технология является полностью открытой, имеет низкий порог вхождения, но в то же время обладает развитой внутренней технологической инфраструктурой, позволяющей строить большие масштабируемые программные решения. Для данной технологической платформы наработано большое количество дополнительных программных продуктов, позволяющих быстро наращивать базовую функциональность до требуемого уровня. Решения, построенные на базе продуктов Python, Zope, Plone, с одной стороны, имеют низкий порог вхождения и позволяют работать с ними пользователям, не имеющим специальной подготовки, с другой стороны, имеют достаточно мощные программные механизмы, позволяющие быстро создавать информационные порталы, реализующие сложные программные модели. Технология Zope Enterprise Objects (ZEO) позволяет создавать сайты, обслуживающие большое количество запросов к ним путем распределения этих запросов по узлам кластера с балансировкой нагрузки.

Основными проблемами являются на данный момент несколько меньшая распространенность данной технологии по сравнению с PHP и меньшая поддержка среди крупных игроков ИТ-рынка, чем у Java технологий, хотя по каждому из указанных направлений видна положительная динамика.

E.5.3.1 Python

Python — это свободный интерпретируемый объектно-ориентированный расширяемый встраиваемый язык программирования сверхвысокого уровня.

Распространяется на условиях лицензии Python license: http://www.python.org/psf/license/

Сайт проекта: http://www.python.org/

E.5.3.2 Zope

Zope — это объектно-ориентированная платформа, сервер приложений, предназначенный для создания динамических web-приложений и интерактивных сайтов.

Распространяется на условиях лицензии Zope Public License: http://www.zope.org/Resources/License/

Сайт проекта: http://www.zope.org/

E.5.3.3 Plone

Plone — профессиональная система управления сайтами (CMS) на основе лицензий с открытым исходным кодом. Plone работает на сервере приложений Zope, в основе которого лежит объектно-ориентированная база данных ZODB. Весь комплекс написан на широко распространенном языке программирования Python. И все компоненты доступны под лицензиями с открытым исходным кодом.

Plone легок в изучении и не требует специальных знаний для начала работы, за что получил множество наград как профессиональное средство для создания корпоративных сайтов. Дистрибутивы Plone доступны для Windows, Linux, BSD, Mac OS X и многих других операционных систем.

Распространяется на условиях лицензии GNU General Public License: http://www.gnu.org/copyleft/gpl.html

Сайт проекта: http://www.plone.org/

Е.6 ГИС

Е.6.1 PostGIS

PostGIS является расширением для объектно-ориентированного сервера реляционных баз данных PostgreSQL

http://www.postgresql.com/

PostGIS добавляет поддержку геоинформационных типов данных для сервера PostgreSQL, который после этого может выступать в качестве spatial backend'а базы данных для ГИС, подобно системе ArcSDE компании ESRI или расширению Spatial СУБД Oracle. PostGIS следует спецификации OpenGIS:

http://www.opengis.org/

и «Simple Features Specification for SQL»:

http://www.opengis.org/docs/99-049.pdf

Распространяется на условиях лицензии GNU General Public License: http://www.gnu.org/copyleft/gpl.html

Сайт проекта: http://postgis.refractions.net/

Е.6.2 MapServer

MapServer — это открытая среда разработки картографических интернет приложений. Данная система позволяет получать отображение геоинформационных данных (карт, растровых изображений и векторных данных) через WEB-интерфейс.

Имеет программные интерфейсы для PHP, Python, Perl, Ruby, Java, и C#.

Имеет возможность работы с различными растровыми и векторными форматами данных:

- TIFF/GeoTIFF, EPPL7 и др. через GDAL:

  http://gdal.maptools.org/formats_list.html

- ESRI shapefiles, PostGIS, ESRI ArcSDE, Oracle Spatial, MySQL и др. через OGR:

  http://ogr.maptools.org/ogr_formats.html

- Open Geospatial Consortium (OGC):

  http://www.opengeospatial.org/

  web-спецификации: WMS (клиент/сервер), не транзакционный WFS (клиент/сервер), WMC, WCS, Filter Encoding, SLD, GML, SOS

Распространяется на условиях лицензии MapServer License: http://mapserver.gis.umn.edu/License

Сайт проекта: http://mapserver.gis.umn.edu/

Е.6.3 GDAL/OGR

GDAL — библиотека для работы с растровыми географическими форматами файлов данных. Как библиотека, GDAL предоставляет вызывающему приложению единую обобщённую модель данных для всех поддерживаемых форматов файлов данных. В состав GDAL входит C++ open source библиотека OGR (http://gdal.maptools.org/ogr), предоставляющая сходные возможности для работы с векторными данными, включая форматы ESRI Shapefiles, S-57, SDTS, PostGIS, Oracle Spatial, Mapinfo mid/mif и TAB.

Распространяется на условиях лицензии X/MIT: http://gdal.maptools.org/faq.html#license

Сайт проекта: http://gdal.maptools.org/

Приложение F

Примеры реализации

В данном разделе будут рассмотрен ряд примеров реализации отдельных модулей ИТ-инфраструктуры на базе ПО с открытым кодом. Материалы раздела основываются на результатах ряда проектов, выполнявшихся компанией «ОАО Линукс Инк»: http://www.linux-ink.ru.

F.1 Специализированный дистрибутив

Одной из уникальных возможностей предлагаемых решений является возможность создания специализированных наборов программного обеспечения, оптимизированных под конкретное аппаратное обеспечение и под тот круг задач, которые они будут решать. Имеющиеся технологии позволяют:

- управлять наборами программных пакетов, включаемыми в дистрибутив, и их группами;
- управлять поведением инсталлятора и программой начального конфигурирования системы;
- настраивать начальный загрузчик системы, см. Рис. F.1;
- задавать стандартные варианты оформления и механизмы поведения системы, см. Рис. F.2;
- задавать стандартные пользовательские профили системы, см. Разд. F.4.

Для каждого специализированного дистрибутива могут быть предусмотрены специальные механизмы развертывания, сопровождения и поддержки (см. Разд. F.7), а также репозитории ПО.

F.2 Модель ЦОД

ЦОД — это вычислительная инфраструктура (набор взаимосвязанных программных и аппаратных компонентов, организационных процедур, мест локации и персонала), предназначенная для безопасной централизованной обработки, хранения и предоставления данных, сервисов, приложений, и обладающая высокой степенью виртуализации своих ресурсов. ЦОД должен поддерживать сервисы авторизации, печати, файловый, удаленного запуска приложений (в том числе унаследованных).

Простейший прототип ЦОД может быть реализован на трех серверах. На первом сервере установлена ОС Scientific Linux 4.1, на втором — Windows 2000 Server. Третий сервер является резервным для первых двух серверов и может работать как под управлением ОС Linux, так и под управлением ОС Windows. Указанные системы также могут выполняться в среде виртуальных машин, что облегчает задачи развертывания, восстановления при сбойных ситуациях и помогает снизить лицензионные выплаты за коммерческое ПО.

Сервер 1, работая с ОС Linux, выполняет следующие задачи:

1. Сервис авторизации.
2. Система электронного документооборота.
3. Информационный портал.
4. Файловый сервис.

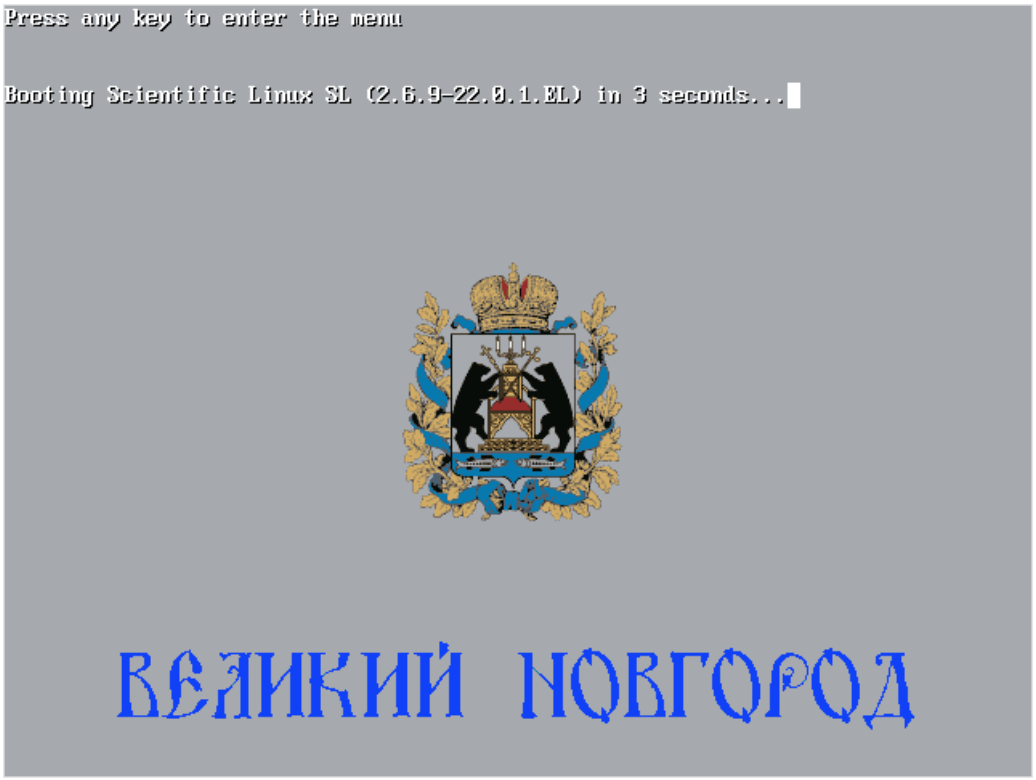

Рис. F.1: Экран начальной загрузки специализированного дистрибутива

5. Сервис печати.

Сервер 2 под управлением ОС Windows предназначен для выполнения следующих задач:

1. Терминальный сервер для работы с унаследованным программным обеспечением.
2. Файловый сервис.
3. Сервис печати.

Сервер 3 является резервным для первого и второго узла. Он может работать как под управлением ОС Linux, так и под управлением ОС Windows, и выполнять задачи, возложенные на любой из первых двух серверов, в случае их отказа. Дисковые устройства основных серверов и резервного сервера объединены в программный RAID 1 через интерфейс iSCSI, что обеспечивает сохранность данных. На Рис. F.4 представлена схема опытного образца ЦОД.

F.3 Прототип РОП

В прототипе была реализована базовая пользовательская среда РОП как унифицированный набор технологий, программных и аппаратных средств, обеспечивающий работу должностных лиц с различными автоматизированными системами и информационными ресурсами региона. Была продемонстрирована возможность проведения настроек пользовательского окружения для выполнения конкретных задач деятельности средствами удаленного доступа.

РОП — это комплекс программных средств и информационных ресурсов, предназначенный для обеспечения повседневного рабочего процесса пользователей по следующим направлениям:

- работа с офисными документами различных типов (текстовыми, электронными таблицами, презентациями и т.п.);
- обеспечение доступа в интернет и интранет (браузер веб-страниц, электронная почта и т.п.);
- запуск и работа с приложениями и ресурсами;

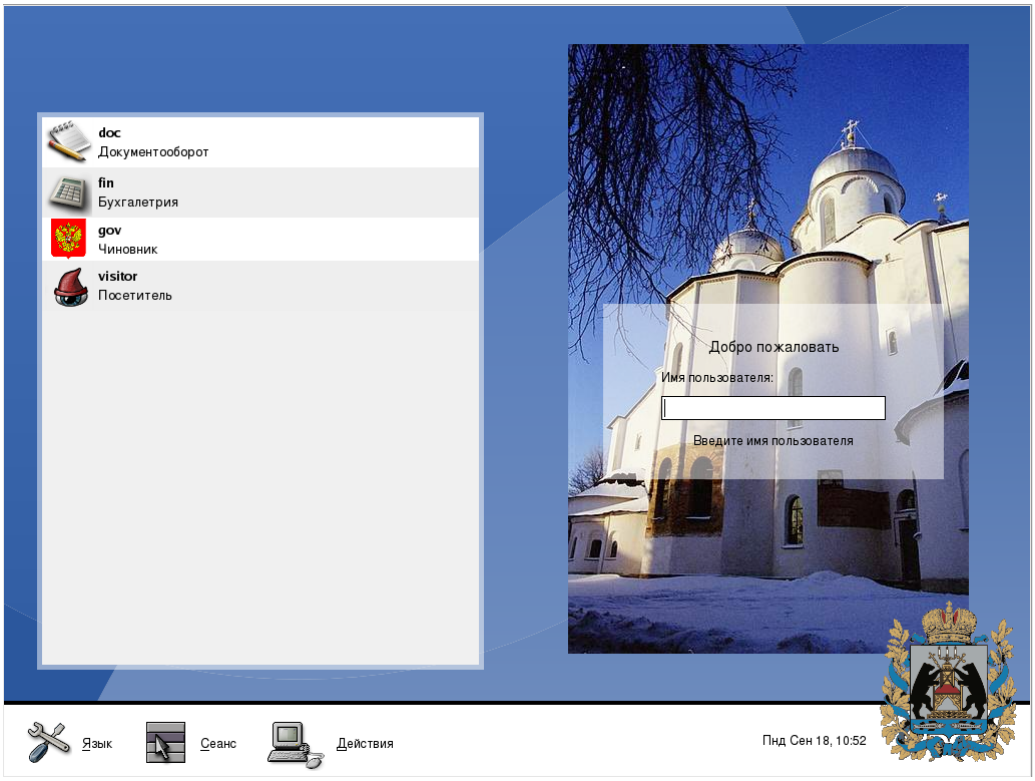

Рис. F.2: Экран входа в систему специализированного дистрибутива

- запуск и работа с унаследованными DOS и Windows приложениями.

Данный комплекс позволяет полноценно работать в среде Интернет/Интранет, предоставляет возможность работать с файлами в формате Microsoft Office и запускать на исполнение унаследованные приложения. Активные элементы, предназначенные для запуска унаследованных приложений, работающих в среде Microsoft Windows Termnail Server (см. Разд. F.2), могут быть интегрированы в окружение рабочего стола пользователя и запускаются в отдельных окнах, при этом возможность доступа к остальному Windows-окружению может быть заблокирована. Таким образом, возможна изоляция Windows-среды исполнения в специальной зоне сети и блокировка потенциально возможных брешей для вирусных атак (Интернет, электронная почта, офисные документы и т.д.), spyware и «программных закладок». При этом функциональные возможности всего комплекса в целом и удобство пользования пользовательским окружением не снижаются.

Общий вывод по результатам демонстрации представленного программного обеспечения РОП:

- продемонстрированные программные средства в полной мере обеспечивают работу должностных лиц в плане осуществления повседневной деятельности с различными автоматизированными системами и информационными ресурсами региона;
- интерфейс программных средств интуитивно понятен для работников среднего уровня компьютерной грамотности и не требует исключительных затрат на переучивание.

Предлагаемые технологии и программные компоненты, лежащие в основе прототипа Инфраструктуры рабочих мест, обладают следующими преимуществами:

- все предлагаемые программные компоненты обладают нулевой стоимостью лицензий на использование;
- в составе предлагаемых решений имеются развитые средства управления (в том числе удаленного) программными компонентами, развернутыми на рабочих станциях;
- возможные варианты модели вычислений позволяют оптимально распределять вычислительную нагрузку между рабочими станциями и серверами, входящими в Опытный образец Центра обработки данных;
- используемая операционная система Linux обеспечивает высокую степень информационной безопасности и надежности;

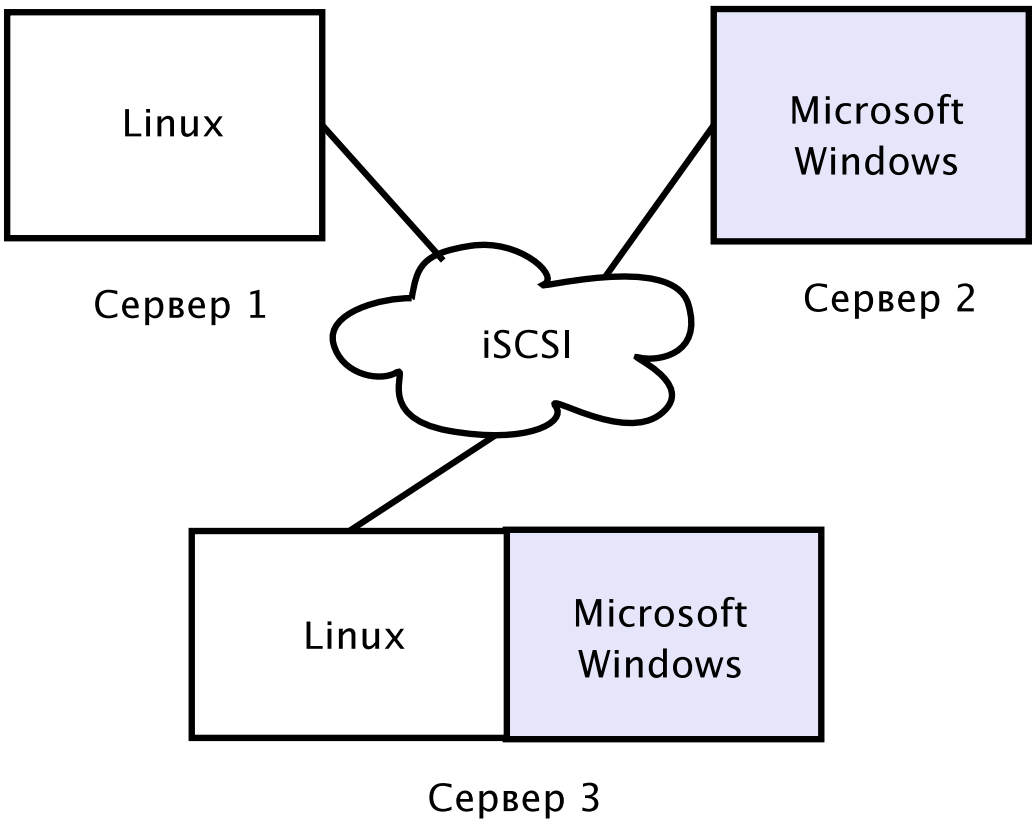

Рис. F.3: Модель ЦОД

- обслуживание рабочих станций, работающих под управлением операционной системы Linux, может выполняться удаленно, в том числе, с использованием низкоскоростных каналов связи.

F.4 Специализированные РОП

На базе ОС Linux возможно построение специализированных программных интерфейсов, настроенных только на выполнение задач работника администрации или внешнего посетителя. Доступ к информационным ресурсам (информационно-правовая база, интеграция с внутренним порталом документооборота) и унаследованным приложениям (Lotus Notes, ГИС Microstation) осуществляется через элементы, интегрированные в окружение рабочего стола, см. Рис. F.5. Возможно создание минимальных пользовательских окружений, где запускается единственное полноэкранное приложение, например веб-броузер. При этом пользователь не имеет возможности даже переместить его или поменять его размеры, выход из приложения автоматически ведет к выходу из пользовательской сессии.

F.5 Интеграционные порталы

Интеграционные порталы должны обеспечивать сопряжение различных компонент региональной ИТ-инфраструктуры. Портал выступает в качестве базового и унифицированного интерфейса, позволяет создать единую точку входа для работы с приложениями с общей политикой авторизации и разграничения прав доступа, единым пользовательским интерфейсом, возможностью регулирования набора предоставляемых сервисов. Функция портала может заключаться также в предоставлении пользовательского интерфейса к компонентам ИТ-инфраструктуры.

Портальные технологии могут быть использованы для решения различных прикладных задач, например:

- Информационный интернет-портал, см. Разд. F.5.2
- СЭД, см. Разд. F.5.3

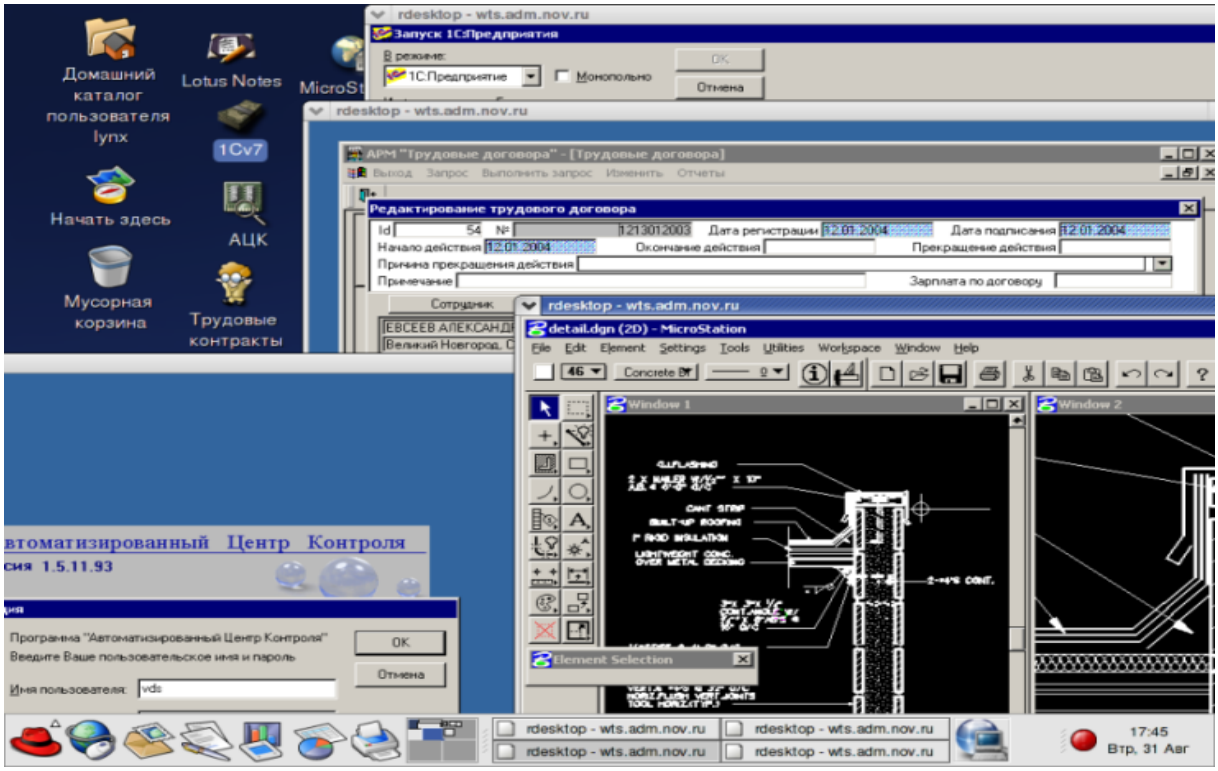

Рис. F.4: Прототип РОП

- Поисково-аналитический ресурс, см. Разд. F.5.4
- Портал службы поддержки, см. Разд. F.7.2

F.5.1 Общая архитектура системы

Применение единой технологической базы для построения портальных решений позволяет упростить решение задач разработки, внедрения и поддержки прикладных решений, а также упростить их интеграцию в единый программный комплекс.

В описываемых реализациях был использован набор технологий Python, Zope, Plone. Такое решение позволяет:

- поддерживать работу нескольких порталов на одном физическом сервере;
- строить достаточно гибкую портальную систему с возможностью распределения нагрузки по нескольким узлам через организацию балансирующих кластеров;
- получать доступ к объектам, содержащимся в портале, из внешних программ, запускаемых с удаленных компьютеров, например, через интерфейс XML-RPC, доступный практически для всех программных сред, см. http://www.xmlrpc.org.

F.5.2 Информационный интернет-портал

Информационный интернет-портал представляет унифицированный пользовательский интерфейс для пользователей различных категорий (граждан, работников администрации и законодательных органов). Основные функции:

- освещение деятельности местных Администрации и Думы путем размещения в нем не только нормативных актов и документов Администрации города и Думы, но и иной официальной информации, как постоянно обновляющейся, так и стабильной (например, история, памятники культуры и достопримечательности и т.п.) и предоставления возможности всем пользователям интернет ознакомиться с этой информацией в удобной для них форме;

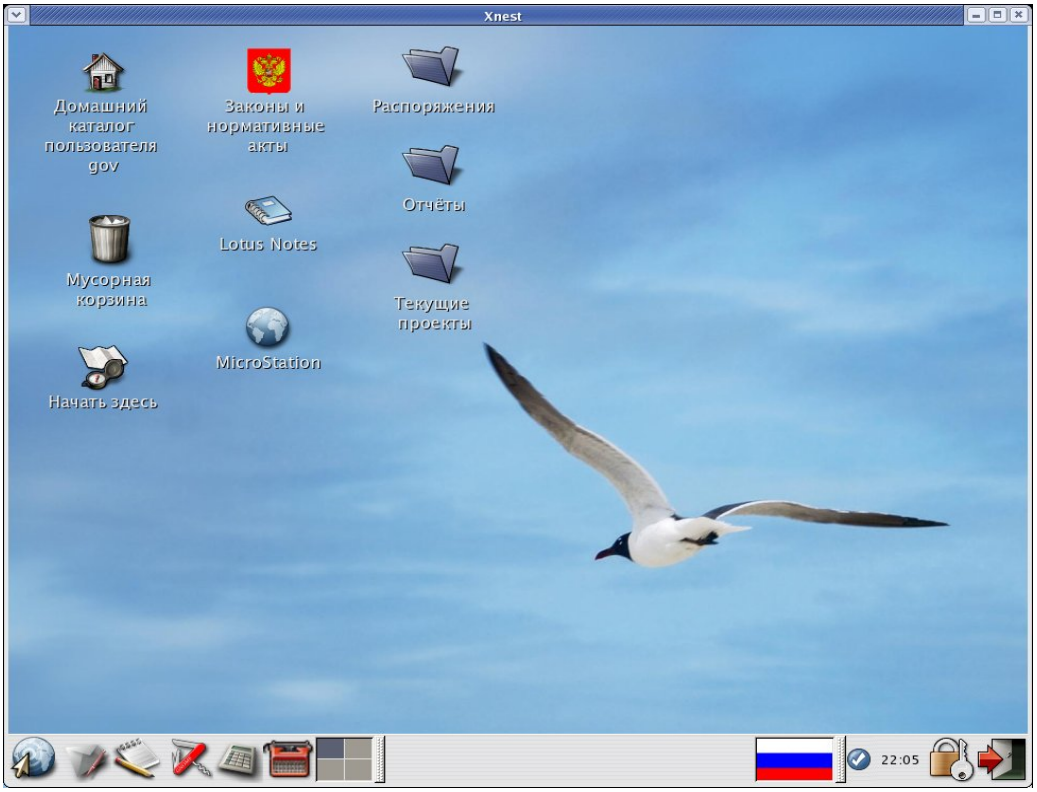

Рис. F.5: Специализированное РОП

- осуществление обратной связи путем предоставления возможности послать письмо (заявление, обращение и т.д.) в Администрацию по электронной почте, при этом обязательным условиям для отправителя письма должно быть сообщение своих данных (фамилии, имени, адреса).

В портале размещаются законодательные акты, решения, постановления, документы и иная информация, предназначенные для широкого информирования граждан. Для структурированного размещения таких документов портал содержит тематические разделы и средства для размещения документов в таких разделах. Для доступа и поиска информационных материалов портал предоставляет удобные, интуитивно понятные средства (поисковые системы, средства навигации и т.д.).

Создаваемые порталы включает в себя программное обеспечение портала для размещения информации и почтовый сервер для обслуживания запросов граждан (SMTP, IMAP). В порталах должны размещаться законодательные акты, решения, постановления, документы и иная информация, предназначенные для широкого информирования граждан. Для структурированного размещения таких документов портал содержит тематические разделы и средства для размещения документов в таких разделах. Для доступа и поиска информационных материалов порталы должен предоставляет удобные, интуитивно понятные средства (поисковые системы, средства навигации и т.д.). Портал обеспечивает возможность оперативного размещения информационных ресурсов работниками администрации и прием обращений граждан.

Размещение новой информации производится через интуитивно-понятный пользовательский интерфейс, представляющий из себя иерархический набор папок с размещенными в них объектами. Папки и элементы уникальным образом отображаются в URL-пространстве сайта.

Стандартный набор типов объектов: документ, гостевая книга, событие, файл, папка, форум, изображение, ссылка, новость, опрос, тема. Набор типов может быть расширен путем установки дополнительных программных пакетов.

В режиме редактирования документов возможно использование WYSIWYG-средств редактирования текстовых полей, наподобие Microsoft Word, см. Рис. F.6. При необходимости из WYSIWYG-режима можно переключиться в непосредственное редактирование HTML-кода страницы.

В ходе размещения материалов в портале действует система разграничения прав доступа и поддерживается система обеспечения прохождения рабочего цикла документа (workflow). Основные виды пользователей:

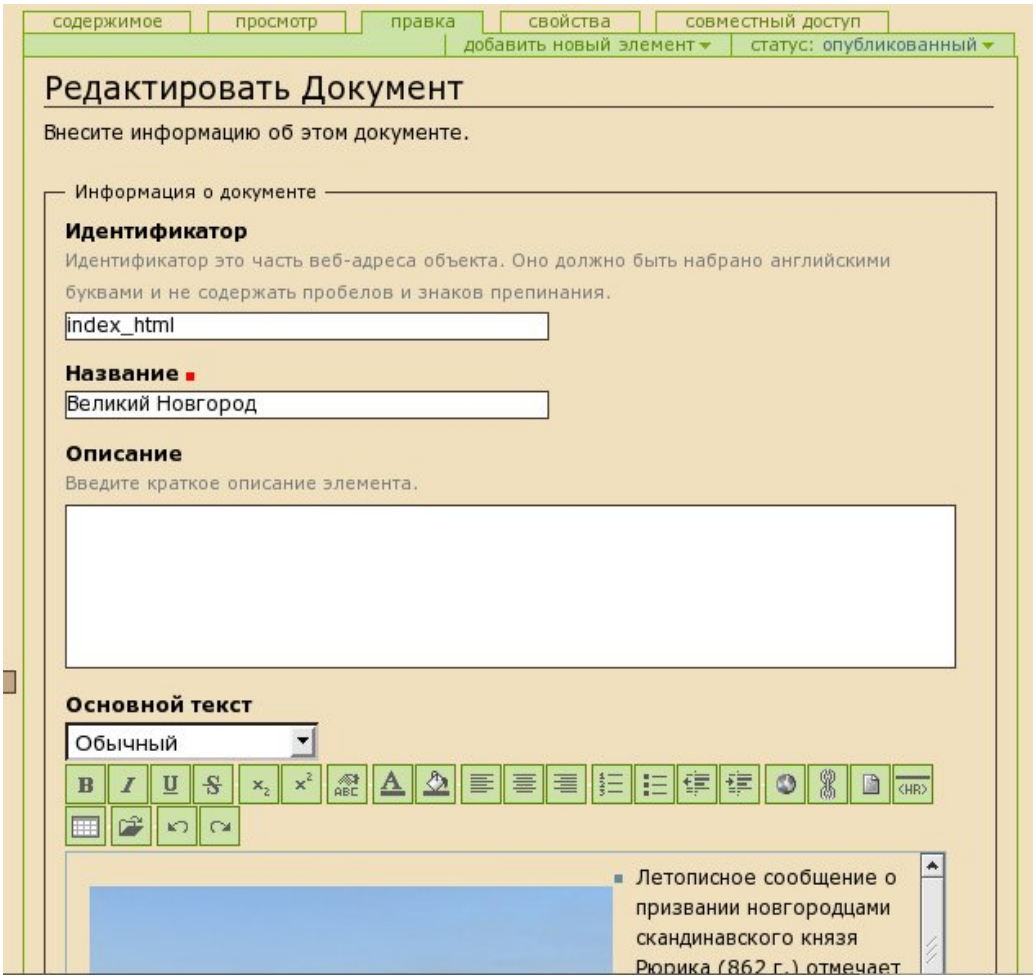

Рис. F.6: Редактирование документа

- анонимные
- авторизованные
- администратор портала

Авторизованные пользователи и их группы, в свою очередь, могут иметь различные роли для различных папок и элементов. В рамках стандартного набора ролей и стандартного workflow возможно:

- корреспонденту — размещение и редактирование документа
- редактору — разрешение на публикацию документа, отмена публикации, отправка документа на доработку

Стандартный набор ролей и workflow может быть изменен администратором портала через веб-интерфейс.

F.5.3 СЭД

Для Центра Информационного обеспечения и услуг был создан прототип системы управления документооборотом.

Система управления документооборотом поддерживает процессы:

- получения/подготовки, регистрации, управления движением и хранением документов;
- управления операциями (регламентами), связанными с обработкой документов должностными лицами и сотрудниками ОМСУ;
- контроля актуального состояния документов и исполнительской дисциплины сотрудников ОМСУ;

- ведения информационной базы документов для быстрого доступа к документам, необходимым как должностным лицам и сотрудникам ОМСУ, так и внешним агентам, включая организации и жителей муниципального образования.

В соответствии с положениями рабочей программы возможности системы управления документооборотом ограничены рамками процесса подготовки и публикации правовых актов Администрации Великого Новгорода. Данный процесс определен специалистами Администрации как наиболее характерный и включающий в себя все необходимые элементы управления регламентами и операциями, а также все необходимые элементы документационного обеспечения управления.

Процесс подготовки и публикации правовых актов включает следующие стадии:

- разработку проекта документа;
- согласование проекта документа;
- утверждение проекта документа;
- регистрацию документа;
- опубликование документа.

В прототипе системы управления документооборотом был реализован процесс прохождения всех стадий подготовки и публикации документа, автоматического изменения статуса документа в зависимости от текущей стадии и рассылки извещений о таких изменениях по электронной почте лицам, ответственным за текущую стадию прохождения документа, разграничения доступа к документу в зависимости от статуса документа.

F.5.4 Сопряжение интеграционного портала и РОП

Данный программный комплекс был разработан в рамках научно-исследовательской программы Министерства Экономического Развития и Торговли «Концепция создания распределенного гетерогенного информационного пространства». Главной целью работ было создание интегрированной информационно-поисковой системы, предоставляющей доступ к широкому набору ресурсов:

- Портал системы управления проектами (МЭРТ) http://projects.economy.gov.ru — HTTP-доступ;
- ИПС «Законодательство России» (ФСО Спецсвязь России) — HTTP-доступ, MySQL;
- Портал электронной библиотеки с возможностью хранения и поиска в массиве полнотекстовых документов (Линукс Инк) — HTTP-доступ, XML-RPC;
- Поисковый библиотечный портал, дающий возможность формирования распределенных поисковых запросов по всем библиотечным ресурсам членов консорциума АРБИКОН (порядка 200 библиотек) — HTTP-доступ, Z39.50
- Портал «Электронная Земля» (ВИНИТИ) — HTTP-доступ, SRU.

В качестве базового решения была выбрана технологическая платформа Plone. Для нее были созданы следующие программные модули:

- электронная библиотека с возможностью хранения и поиска в массиве полнотекстовых документов;
- поисковые механизмы, дающие возможность поиска как по метаданным, так и по полнотекстовым массивам информации с учетом русской морфологии;
- модули сопряжения с внешними информационными источниками;
- модули сопряжения с программными компонентами РОП.

Одной из задач, решавшихся в данном проекте, являлась демонстрация возможности интеграции данного портального решения с рабочим окружением пользователя. Были продемонстрированы возможности сопряжения со следующими компонентами РОП (см. Рис. F.7):

- FireFox — стандартный доступ через веб-интерфейс с возможностями поиска (универсального полнотекстового и специализированного для данного типа информационного источника с учетом метаданных), отображение данных (метаданные и полнотекстовые документы, представленные в различных форматах) и их редактирование (для авторизованных пользователей);

- OpenOffice.org — формирование поискового запроса в интеграционный портал и автоматическая генерация аналитической подборки материалов, соответствующих данному запросу;
- GNOME Desktop — включение в рабочий стол пользователя активного элемента (desklet'а), осуществляющего постоянный мониторинг состояния информационных ресурсов по заданному в конфигурации поисковому запросу к порталу.

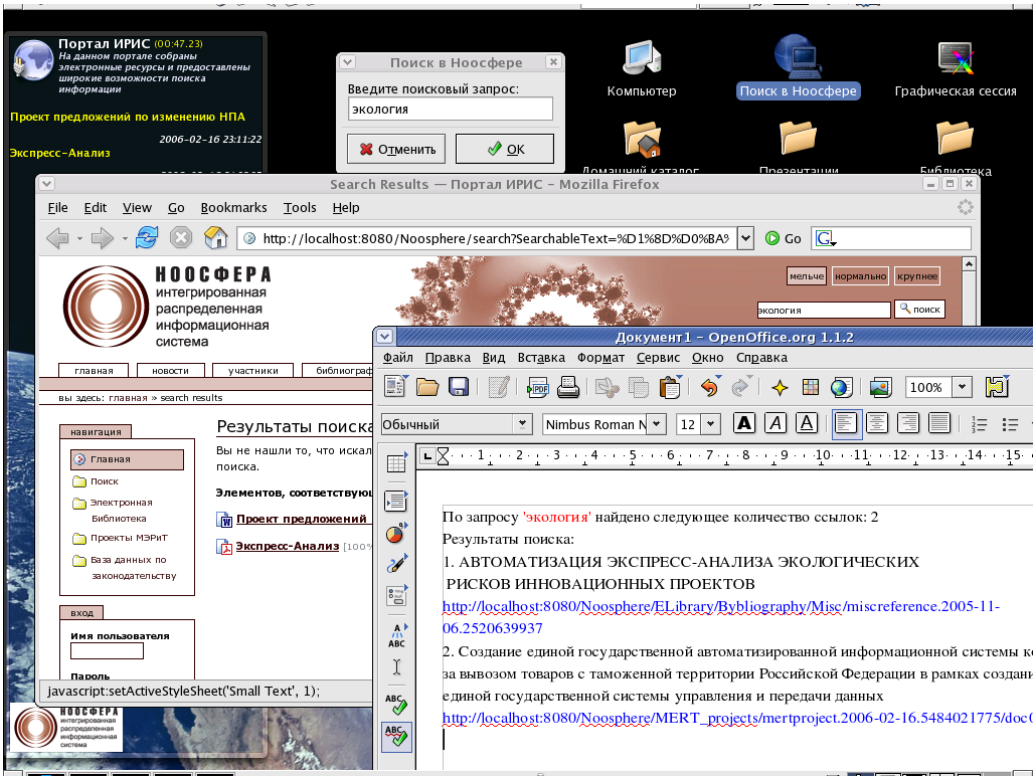

Рис. F.7: Взаимодействие различных элементов пользовательского окружения (Firefox, GNOME Desktop, OpenOffice.org) с порталом

F.6 ГИС

В этой работе была реализована технологическая платформа конвертации и отображения картографических данных, был создан WEB-модуль публикации электронных карт, способный обеспечить выполнение набора базовых функций, необходимых для работы с картографической информацией, представленной в электронном виде, и возможность импорта в такую систему имеющихся геоинформационных данных.

F.6.1 Общая архитектура системы

В качестве основы системы публикации использовались свободно распространяемые компоненты, построенные в соответствии с принципами программного обеспечения с открытым кодом (Open Source). Такое решение имеет существенные преимущества перед традиционно использовавшимися закрытыми коммерческими системами, поскольку снижает порог вхождения и стоимость владения, позволяет создать более защищенную и устойчивую к внешним воздействиям систему с высокой степенью отчуждаемости полученного решения.

В результате проведенного анализа существующих программных проектов были выбраны следующие программные продукты:

- система хранения — PostGIS;
- модули для сопряжения геоинформационных данных с общим информационным пространством портала через WEB-интерфейс — MapServer;
- программы для конвертации существующих данных — GDAL/OGR.

F.6.2 Результаты

Нам был предоставлен набор тестовых данных в виде файлов формата «ESRI Shapefile» и «MapInfo TAB». В тестовый набор были включены данные по расположению водных ресурсов, лесонасаждений, районов, кварталов и домов.

Были выполнены следующие работы:

1. Были осуществлен подбор требуемого программного обеспечения и предварительная сборка и установка программных пакетов на дистрибутив Scientific Linux 4x.
2. При помощи программных модулей ogr2ogr, входящих в программный пакет gdal, на языке программирования Python был создан конвертор для импортирования предоставленных тестовых данных в базу данных PostgreSQL с расширением PostGIS. Для реализации возможности отслеживания изменений в массиве картографических данных с течением времени были включены функции timetravel PostgreSQL.
3. Было осуществлено конфигурирование MapServer'а и привязка слоёв к соответствующим запросам к базе данных.

Результатом работ явился полный перенос предоставленных данных и создание WEB-интерфейса для их отображения.

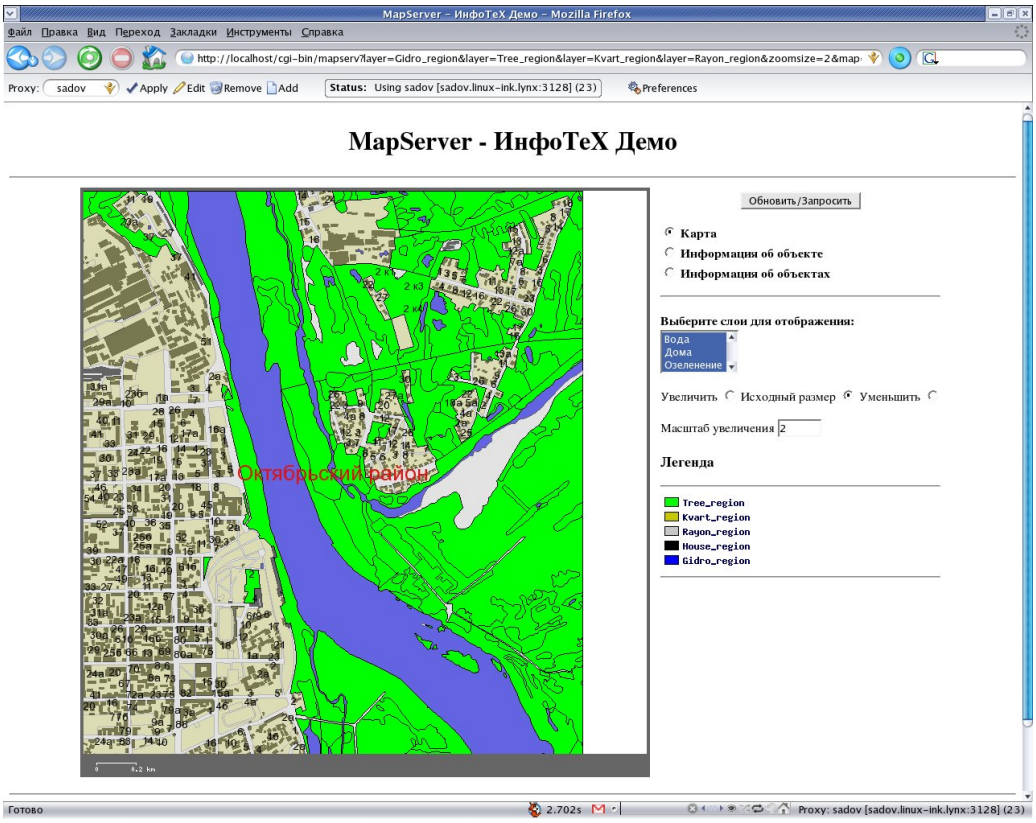

Рис. F.8: Веб-интерфейс для доступа к картографической информации (ГИС)

F.7 Автоматизация службы поддержки и сопровождения

F.7.1 Развертывание и тестирование ПО

Данный программный модуль позволяет проводить инсталляцию специализированного набора ПО на большое количество компьютеров, и проведение их тестирования в автоматическом режиме. Компоненты, составляющие данный модуль, могут использоваться в технологических центрах различного уровня, в том числе в местных территориальных отделениях.

Пример реализации: учебный центр LEC. В ходе подготовки к учебным курсам постоянно возникает задача оперативной установки требуемого программного обеспечения. Для этого была разработана специализированная система, позволяющая снимать образы дисков или их разделов, на которых установлен требуемый набор ПО. Весь процесс установки заключается в выборе администратором системы требуемых для установки образов и загрузки рабочих станций по PXE. Дальнейшие действия по установке образа на диск и тестово-настроечные процедуры выполняются в автоматическом режиме.

F.7.1.1 Клиентская часть

В качестве клиентской части для администратора системы предлагается использовать компьютер под управлением ОС Linux с базовым набором утилит и набором специализированного ПО для облегчения задач администрирования процесса инсталляции и тестирования.

F.7.1.2 Серверная часть

В качестве сервера предлагается использовать компьютер под управлением ОС Linux с поддержкой следующих сервисов:

1. DHCP-сервер для динамического выделения IP-адресов клиентским машинам;
2. PXE-сервер для обеспечения возможности удаленной загрузки, например, бездисковых рабочих станций, работающих в режиме «тонкий клиент»;
3. Файловый сервер с поддержкой протокола NFS.

F.7.2 Портал службы поддержки и сопровождения

В рамках данного модуля предлагается создание системы удаленной поддержки работы территориально удаленных пунктов интернет-доступа. Предлагаемые решения на базе Open Source позволяют проводить большую часть работ по их поддержке удаленно, в том числе на каналах связи, обладающих низкой пропускной способностью.

Система поддержки оптимизирована для работы через WEB-браузер. Поддерживается процедура авторизации пользователей и осуществляется разграничение доступа к различным сервисам. Возможно разделение инженеров на различные сервисные группы с перенаправлением на них указанных видов запросов, полнотекстовый поиск по накопленной базе заявок и решений, получение отчетов о состоянии заявок необходимой формы начальником службы и т.д.

Информация о системных и аппаратных сбоях собирается в системных журналах и отправляется в соответствующую службу поддержки автоматически.

Служба установки обновлений также работает в автоматическом режиме. Набор необходимых для обновления пакетов размещается на специализированном сервере службы технической поддержки, откуда соответствующие пакеты устанавливаются на компьютеры территориальных пунктов доступа.

Возможно создание территориально-распределенной сети центров поддержки, включающей в себя:

- региональные центры обслуживания;
- централизованную систему обслуживания заявок (Service Desk).

Процедура обслуживания предполагает:

1. размещение заявки на обслуживание сотрудниками пункта доступа в централизованном ресурсе технической поддержки;
2. выделение ответственного инженера в региональном центре поддержки для оказания поддержки первого уровня;
3. возможность эскалации проблемы в службы поддержки более высокого уровня, вплоть до служб поддержки компании разработчика;

F.7.2.1 Клиентская часть

В качестве клиентской части для инженеров и операторов службы технической поддержки предлагается использовать компьютеры под управлением ОС Linux с базовым набором утилит и стандартным WEB-браузером. С рабочих мест инженеров должна быть обеспечена возможность авторизованного удаленного входа на удаленные станции, входящие в инфраструктуру пунктов доступа и попадающие в его сферу ответственности.

F.7.2.2 Серверная часть

В качестве сервера предлагается использовать компьютеры под управлением ОС Linux с установленным на них портальным ПО (Plone) и специализированными пакетами для создания порталов технической поддержки.

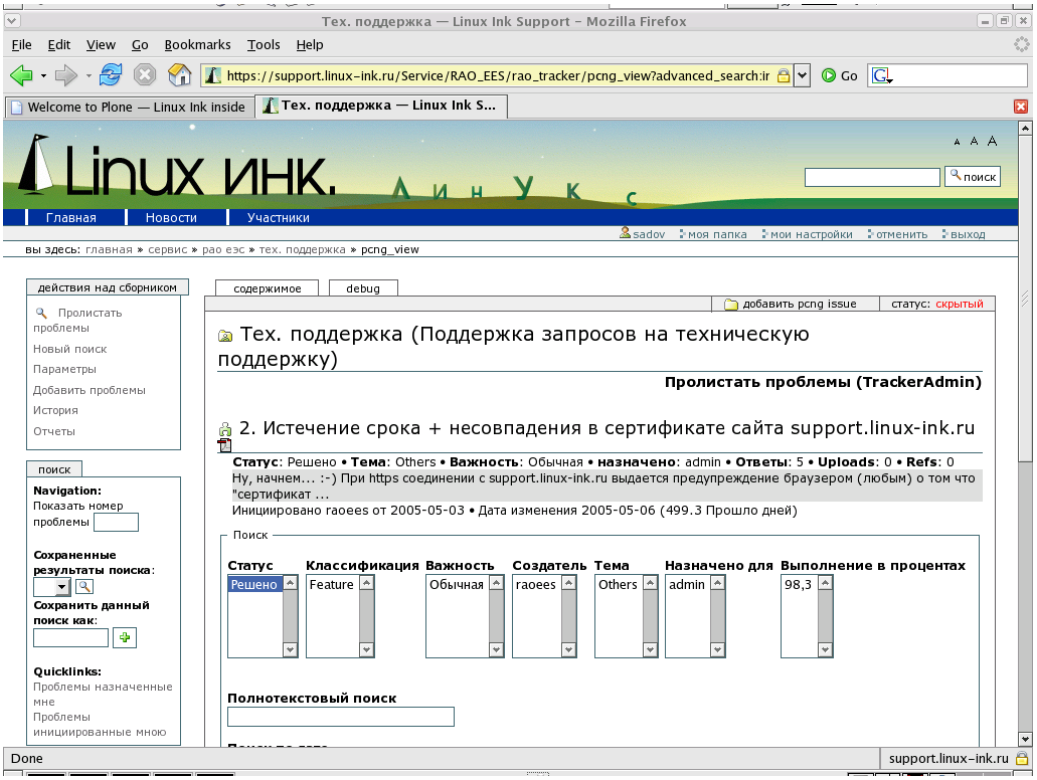

Рис. F.9: Портал службы технической поддержки